Institut für Formale Methoden der Informatik
Abteilung Formale Konzepte
Universität Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Studienarbeit Nr. 2257

Algebraische Hilfsmittel für die Aneinanderreihung von Nebenläufigen Systemen

Bastian Bischoff

Studiengang: Informatik

Prüfer: Prof. Dr. Volker Diekert

Betreuer: Dr. Manfred Kufleitner

begonnen am: 20. Januar 2010

beendet am: 20. April 2010

CR-Klassifikation: F.4.3

Inhaltsverzeichnis

1 Einleitung **5**

2 Grundlagen **7**

2.1 Das Schützenberger-Produkt endlicher Monoide 7

2.2 Spurmonoide 8

2.3 Unabhängigkeitsmonoide (I-Monoide) 8

2.4 Skeletonmonoide 9

3 Das Schützenberger-Produkt endlicher I-Monoide **13**

3.1 Definition 13

3.2 Spursprachen des Schützenberger-Produkts 15

4 Anwendungen **25**

4.1 Schützenbergers Theorem für sternfreie Sprachen 26

4.2 Erkennbare Ausdrücke 29

5 Zusammenfassung **31**

1 Einleitung

Eine mögliche Beschreibung von sequentiellen Systemen, ist die Darstellung mit formalen Wortsprachen. Prozesse entsprechen dabei Zeichen, eine Abfolge von Prozessen einem Wort. Zur Untersuchung von formalen Wortsprachen wurden bereits viele Anstrengungen unternommen und es stehen zahlreiche Werkzeuge zur Verfügung. Dem hintereinander Ausführen von mehreren sequentiellen Systemen entspricht das Produkt von Sprachen. Jedoch existiert keine einfache Operation auf der Seite der Monoide, die dem Produkt von Sprachen entspricht. Schützenberger hat in [17] die Konstruktion eines solchen Produkts angegeben. Das *Schützenberger-Produkt* von Monoiden stellt inzwischen eines der wichtigsten Werkzeuge dar um Probleme zu untersuchen, die das Produkt von Sprachen betreffen. Wir beschäftigen uns in dieser Arbeit nur mit regulären Sprachen. Diese entsprechen genau den von endlichen Monoiden erkannten Sprachen [4, 2. Kapitel]. Liegen reguläre Sprachen sowie endliche Monoide vor, welche die Sprachen erkennen, so erkennt das Schützenberger-Produkt der Monoide, das (markierte) Produkt der Sprachen.

In der heutigen Zeit gewinnen nebenläufige Systeme immer mehr an Bedeutung. Jedoch eignen sich formale Sprachen über Wörtern nicht zur Beschreibung dieser Systeme, da die Reihenfolge beim Ablauf von Prozessen in nebenläufigen Systemen nicht immer fest vorgegeben ist. Hierzu wurde 1977 von Mazurkiewicz das Konzept der *Spuren* entwickelt [11]. Im Gegensatz zu Wortsprachen, war der algebraische Ansatz bei Spursprachen bisher nicht ähnlich erfolgreich. Ein möglicher Grund hierfür ist die ungenügende Struktur von Monoiden zur Darstellung der Vertauschbarkeit von Prozessen. Kufleitner und Weil haben kürzlich *Unabhängigkeitsmonoide* entwickelt [10]. Dies sind Monoide, versehen mit einer binären Relation zur Beschreibung von Abhängigkeit und Unabhängigkeit von Prozessen. Unabhängigkeitsmonoide eignen sich für Spursprachen wesentlich besser als algebraische Grundstruktur.

Kapitel 2 enthält grundlegende Defitionen und Erklärungen. Kapitel 3 stellt das Hauptresultat der vorliegenden Arbeit dar. Das Schützenberger-Produkt wird darin auf Unabhängigkeitsmonoide erweitert. Analog zum Schützenberger-Produkt für Wortsprachen erkennt die Erweiterung genau das Produkt von Spursprachen. Damit wurde ein wichtiges Werkzeug bei Wortsprachen auf Spursprachen übertragen. Im 4. Kapitel werden Anwendungen für das Schützenberger-Produkt aufgezeigt. Der ebenfalls von Schützenberger stammende Satz zu sternfreien Sprachen besagt, dass sternfreie Sprachen genau den Sprachen entsprechen, die von endlichen, aperiodischen Monoiden erkannt werden [17, 15]. Mit Hilfe des erweiterten Schützenberger-Produkts wird ein vereinfachter, vollständiger Beweis der Aussage angegeben. Des Weiteren wird ein vereinfachter Beweis zu

folgender Aussage angegeben: Jede zu einem erkennbaren Ausdruck gehörige Sprache, wird von einem endlichen Monoid erkannt. Die Umkehrung dieser Aussage gilt ebenfalls und ist als Satz von Ochmanski bekannt. Abschließend enthält Kapitel 5 eine Zusammenfassung der Ergebnisse dieser Arbeit.
Die vorliegende Studienarbeit wurde in Zusammenarbeit mit Manfred Kufleitner erstellt.

2 Grundlagen

Eine formale Sprache L, ist eine Menge von Wörtern über einem endlichen Alphabet Σ. Automaten und Grammatiken sind bekannte Möglichkeiten, um solche Sprachen zu beschreiben. Eine weniger bekannte Möglichkeit, ist das Erkennen von Sprachen mit Monoiden. Eine Sprache L wird genau dann von einem Monoid M erkannt, wenn ein Homomorphismus $\varphi : \Sigma^* \to M$ existiert, so dass $L = \varphi^{-1}\varphi(L)$ gilt. Wir sagen L wird von M erkannt.

2.1 Das Schützenberger-Produkt endlicher Monoide

Ausgangspunkt seien $\ell+1$ Sprachen $L_0, \ldots, L_\ell$. Für alle $0 \leq i \leq \ell$ ist L_i eine Sprache, die von einem endlichen Monoid M_i erkannt wird. Das Ziel des *Schützenberger-Produkts* ist es, das *markierte Produkt* der Sprachen zu erkennen. Die genaue Aussage von Schützenberger, Straubinger, Reutenauer und Pin [17, 19, 16, 14] lautet

Satz 2.1 *Sei Σ ein endliches Alphabet und seien $M_0, \ldots, M_\ell$ endliche Monoide. Dann erkennt das* Schützenberger-Produkt *$\diamond(M_0, \ldots, M_\ell)$ eine Sprache $L \subseteq \Sigma^*$ genau dann, wenn L als boolsche Kombination von Sprachen der Form*

$$L_{i_0} a_1 L_{i_1} \ldots a_r L_{i_r}$$

dargestellt werden kann. Dabei ist $0 \leq r \leq \ell$, $0 \leq i_0 < i_1 < \ldots < i_r \leq \ell$, $a_k \in \Sigma$ für alle $1 \leq k \leq r$ und $L_{i_j} \subseteq \Sigma^*$ ist eine von M_{i_j} erkannte Sprache ($0 \leq j \leq r$)

Das Schützenberger-Produkt $\diamond$ ist eine Menge von $(\ell+1) \times (\ell+1)$-Matrizen $P = (P_{ij})$. Einträge der Matrizen sind Teilmengen von $M_0 \times \ldots \times M_\ell$. Eine Matrix P_{ij} ist genau dann im Schützenberger-Produkt $\diamond$ enthalten, wenn gilt

- $P_{ij} = \emptyset$ für alle $0 \leq j < i \leq \ell$
- $P_{ij} \subseteq 1 \times \ldots \times 1 \times M_i \times \ldots \times M_j \times 1 \times \ldots \times 1$ für alle $0 \leq i \leq j \leq \ell$
- P_{ii} ist einelementig für alle $0 \leq i \leq \ell$

Ein zu früheren Versionen stark vereinfachter Beweis der Aussage ist in [18] von Simon zu finden. Die Struktur dieses Beweises dient als Grundlage in Kapitel 3.

2.2 Spurmonoide

Sprachen über Wörtern eignen sich zur Darstellung von sequentiellen Systemen. Ein Zeichen entspricht dabei beispielsweise einem Prozess, ein Wort stellt eine Ablauffolge von Prozessen dar. Bei nebenläufigen Systemen ist die Reihenfolge von zwei Prozessen a und b nicht immer fest vorgegeben. So können a und b Prozesse sein, die unabhängig voneinander bearbeitet werden können. In diesem Fall muss zwischen ab und ba nicht unterschieden werden. Diese Vertauschbarkeit trifft aber natürlich nicht auf alle Prozesse zu. Beispielsweise für Initialisierungen oder Synchronisation ist die Reihenfolge der Prozesse wichtig. Als Erweiterung von Wörtern entwickelte Mazurkiewicz 1977 das Konzept von *Spuren* [11]. Spuren berücksichtigen genau diese Vertauschbarkeit mancher Prozesse. Mit $I \subseteq \Sigma \times \Sigma$ bezeichnen wir eine symmetrische, irreflexive Relation über dem Alphabet Σ, die sogenannte Unabhängigkeitsrelation. Es soll $(a, b) \in I$ genau dann gelten, wenn a und b vertauschbar sind. Gilt $(a, b) \notin I$, so sagen wir a und b sind abhängig. Das Paar (Σ, I) bezeichnen wir als Unabhängigkeitsalphabet. Die Unabhängigkeitsrelation I induziert eine Äquivalenzrelation $\sim_I$ über Σ^*. Zwei Wörter u und v sind genau dann äquivalent, wenn wir u durch endlich oftes Vertauschen unabhängiger Zeichen in v überführen können. Formal ist also $u \sim_I v$ genau dann, wenn für ein k Wörter $u = w_1, w_2, \ldots, w_k = v$ existieren, so dass für alle $1 \leq i \leq k$ gilt:

$$\begin{aligned} \exists z_i', z_i'' \in \Sigma^* \; \exists a_i, b_i \in \Sigma : \quad & z_i = z_i' a_i b_i z_i'' \\ & z_{i+1} = z_i' b_i a_i z_i'' \text{ mit } (a_i, b_i) \in I \end{aligned}$$

Wir erhalten also z_{i+1} aus z_i durch Vertauschen der unabhängigen Buchstaben a_i und b_i. Die Äquivalenzklassen der Relation $\sim_I$ heißen *Spuren*. Eine Spur s ist damit eine Klasse von Wörtern, die durch Vertauschen unabhängiger Buchstaben ineinander überführt werden können. Das Monoid $\Sigma^* / \sim_I \; = \; \Sigma^* / \{ u, v \in \Sigma^* : uv = vu \mid (u, v) \in I \}$ bezeichen wir als Spurmonoid $\mathbb{M}(\Sigma, I)$. Eine Spurprache über dem Spurmonoid ist eine Menge von Spuren. Verkürzend schreiben wir $\mathbb{M}$ statt $\mathbb{M}(\Sigma, I)$, falls Σ und I klar sind.

2.3 Unabhängigkeitsmonoide (I-Monoide)

Sprachen über Wörtern standen zur Beschreibung Monoide als mathematische Struktur gegenüber. Für Spursprachen hat sich diese Darstellung als nicht besonders effektiv erwiesen. Daher entwickelten Kufleitner und Weil 2007 sogenannte *Unabhängigkeitsmonoide* (*I-Monoide*) [10]. Unabhängigkeitsmonoide stellen eine Erweiterung von Monoiden dar. Eigenschaften von Spurmonoiden wie Abhängikeiten und Unabhängigkeiten von Zeichen können in Unabhängigkeitsmonoiden, im Gegensatz zu Monoiden, direkt repräsentiert werden. Dafür wird zunächst eine symmetrische Relation $I \subseteq M \times M$ über einem Monoid M benötigt. Die Relation I muss für alle $x, y, z \in M$ folgende Bedingungen erfüllen

- $(x, y) \in I \Rightarrow xy = yx$ in M
- $(x, yz) \in I \Leftrightarrow (x, y) \in I$ und $(x, z) \in I$

- $(x, x) \in I \Leftrightarrow x = 1$
- $(x, 1) \in I$

Sind die Bedingungen erfüllt, so bezeichnen wir I als Unabhängigkeitsrelation über M. Aus der zweiten Bedingung folgt, dass falls $(x, y) \notin I$, auch $(z_1xz_2, z_3yz_4) \notin I$ für alle $z_1, z_2, z_3, z_4 \in M$ gilt. Abhängigkeiten bleiben also auch nach dem Heranmultiplizieren weiterer Elemente erhalten. Diese Aussage werden wir im weiteren Verlauf der Arbeit mehrfach verwenden.
Das Paar (M, I) bezeichnen wir als *Unabhängigkeitsmonoid* (*I-Monoid*). Eine Abbildung $\varphi : (M, I) \to (N, J)$ zwischen zwei I-Monoiden heißt *I-Homomorphismus*, wenn $\varphi : M \to N$ ein Homomorphismus ist und für alle $x, y \in M$ aus $(x, y) \in I$ folgt, dass $(\varphi(x), \varphi(y)) \in J$ gilt. Ein I-Homomorphismus heißt stark, falls $(x, y) \in I$ genau dann gilt, wenn $(\varphi(x), \varphi(y)) \in J$. In dieser Arbeit betrachten wir nur starke I-Homomorphismen. Ein Spurmonoid $\mathbb{M}(\Sigma, I)$ lässt sich als Unabhängigkeitsmonoid auffassen. Es muss lediglich die Unabhängigkeitsrelation von Zeichen auf Spuren erweitert werden. Für zwei Spuren s, t gilt $(s, t) \in I$ genau dann, wenn $alph(s) \times alph(t) \subseteq I$. Dabei ist $alph(s)$ die Menge der Zeichen, die in s vorkommen. Sprachen über Σ^* konnten durch Homomorphismen $\varphi : \Sigma^* \to M$ für ein Monoid M beschrieben werden. Als Gegenstück für Spuren haben wir nun

$$\varphi : \mathbb{M}(\Sigma, I) \to (M, J)$$

für ein I-Monoid (M, J) und einen I-Homomorphismus φ. Da wir nur starke I-Homomorphismen betrachten, wird J von I induziert. Wir sprechen daher im Folgenden sowohl auf $\mathbb{M}$, als auch auf M nur von der Unabhängigkeitsrelation I. Analog zu Sprachen über Wörtern sagen wir, eine Spursprache $L \subseteq \mathbb{M}(\Sigma, I)$ wird genau dann von M erkannt, wenn ein I-Homomorphismus $\varphi : \mathbb{M}(\Sigma, I) \to (M, I)$ mit $\varphi^{-1}\varphi(L) = L$ existiert.

2.4 Skeletonmonoide

Gegeben sei ein Unabhängigkeitsalphabet (Σ, I). Wir sagen ein I-Monoid (M, J) wird stark von (Σ, I) erzeugt, wenn ein Homomorphismus $\varphi : \Sigma \to M$ existiert, so dass M von $\varphi(\Sigma)$ erzeugt wird und φ stark ist, also dass gilt

$$(a, b) \in I \Leftrightarrow (\varphi(a), \varphi(b)) \in J$$

In [10] ist die Konstruktion eines kleinsten Monoids $\mathbb{S}(\Sigma, I)$ angegeben, das stark von (Σ, I) erzeugt wird. Dabei bedeutet klein, dass zu jedem stark von (Σ, I) erzeugten I-Monoid (M, I) eindeutig ein starker surjektiver I-Homomorphismus $\tilde{\varphi} : (M, I) \twoheadrightarrow \mathbb{S}(\Sigma, I)$ existiert.

$$\begin{tikzcd}
 & (\Sigma, I) \arrow[dl, "\varphi"'] \arrow[dr, "\pi"] & \\
(M, I) \arrow[rr, dashed, two heads, "\exists!\tilde{\varphi}"] & & \mathbb{S}(\Sigma, I)
\end{tikzcd}$$

Um $\mathbb{S}(\Sigma, I)$ zu erhalten, wird die Skeleton Konstruktion aus der Graphentheorie verwendet [1, 9]. Man erhält $\mathbb{S}(\Sigma, I)$ als Quotienten der Äquivalenzrelation $\sim_I$, dabei ist

$$a \sim_I b \;\Leftrightarrow\; (\forall c \in \Sigma : (a, c) \in I \Leftrightarrow (b, c) \in I)$$

Es werden also Zeichen zusammengefasst, die sich bezüglich der Unabhängigkeit gleich verhalten. Dass dies das gesuchte Monoid ergibt, so wie ein Beweis zur Existenz und Eindeutigkeit von $\tilde{\varphi}$, kann in [10] nachgelesen werden. In dieser Arbeit wird das Skeleton Monoid verwendet, um Zeichen aus unterschiedlichen I-Monoiden auf Unabhängigkeit zu überprüfen. Dabei ist ein Spurmonoid $\mathbb{M}(\Sigma, I)$ zu einem Unabhängigkeitsalphabet (Σ, I) gegeben. Das Spurmonoid $\mathbb{M}(\Sigma, I)$ ist I-Monoid mit der natürlichen Einbettung $\varphi : (\Sigma, I) \to \mathbb{M}(\Sigma, I)$. Nun lässt sich obige Konstruktion anwenden. Statt $\tilde{\varphi}$ bezeichnen wir diesen starken I-Homomorphismus ebenfalls mit $\mathbb{S}$.

$$\begin{tikzcd}
 & (\Sigma, I) \arrow[dl, "\varphi"'] \arrow[dr, "\pi"] & \\
\mathbb{M}(\Sigma, I) \arrow[rr, dashed, two heads, "\exists!\mathbb{S}"] & & \mathbb{S}(\Sigma, I)
\end{tikzcd}$$

Wir wollen später Zeichen aus $\mathbb{M}(\Sigma, I)$ mit Zeichen aus weiteren I-Monoiden vergleichen. Dies ist auch möglich wenn $\varphi' : \Sigma \to M_1$ nicht bekannt ist, dafür aber der starke I-Homomorphismus $\mathbb{S}_1$ aus unten stehendem Diagramm.

$$\begin{tikzcd}
(\Sigma, I) \arrow[r, dashed, "\varphi'"] \arrow[d, "\varphi"'] \arrow[dr, "\pi"] & (M_1, I) \arrow[d, "\mathbb{S}_1"] \\
\mathbb{M}(\Sigma, I) \arrow[r, two heads, "\mathbb{S}"] & \mathbb{S}(\Sigma, I)
\end{tikzcd}$$

Statt $(\mathbb{S}(a), \mathbb{S}_1(b)) \in I$ für $a \in \mathbb{M}(\Sigma, I)$ und $b \in (M_1, I)$, schreiben wir einfach $(a, b) \in I$. Seien nun mehrere endliche Monoide $M_1, \ldots, M_\ell$, sowie starke I-Homomorphismen $\mathbb{S}_1, \ldots, \mathbb{S}_\ell$ mit $\mathbb{S}_i : M_i \to \mathbb{S}(\Sigma, I)$ gegeben, $1 \leq i \leq \ell$. Daraus ergeben sich Unabhängigkeitsrelationen auf den Monoiden $M_1, \ldots, M_\ell$, wodurch diese zu I-Monoiden werden.

$$\begin{tikzcd}
 & & (M_1, I) \arrow[dl, "\mathbb{S}_1"'] \\
 & & (M_2, I) \arrow[dl, "\mathbb{S}_2"] \\
\mathbb{M}(\Sigma, I) \arrow[r, "\mathbb{S}"] & \mathbb{S}(\Sigma, I) \quad \vdots & \vdots \\
 & & (M_\ell, I) \arrow[ul, "\mathbb{S}_\ell"]
\end{tikzcd}$$

Ist klar welche Abbildung $\mathbb{S}_i$ gemeint ist, so schreiben wir zur Vereinfachung im Folgenden einfach $\mathbb{S}$. Mit Hilfe der Skeleton Abbildungen können wir nun Elemente aus $M_1, M_2, \ldots, M_\ell$ sowie aus $\mathbb{M}$ paarweise auf Unabhängigkeit prüfen. Es ist $(x_i, x_j) \in I$ für $x_i \in M_i$ und $x_j \in M_j$ mit $1 \leq i, j \leq \ell$ genau dann, wenn $(\mathbb{S}(x_i), \mathbb{S}(x_j)) \in I$ im Skeletonmonoid $\mathbb{S}$ gilt. Analog ist $(s, x_i) \in I$ für $s \in \mathbb{M}$ und $x_i \in M_i$ genau dann, wenn $(\mathbb{S}(s), \mathbb{S}(x_i)) \in I$.

3 Das Schützenberger-Produkt endlicher I-Monoide

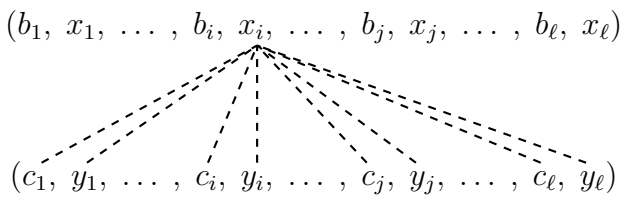
Wir wollen nun das Konzept des Schützenberger-Produkts endlicher Monoide auf I-Monoide erweitern. Wir erhalten dadurch ein Werkzeug, um das markierte Produkt von Spursprachen zu beschreiben.

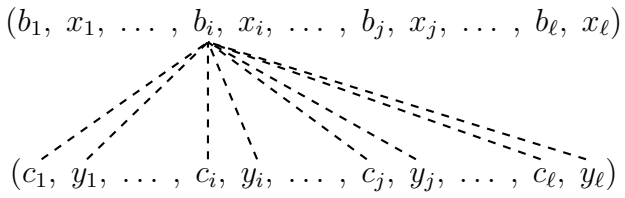
3.1 Definition

Um das Schützenberger-Produkt für Spursprachen zu definieren, benötigen wir zunächst einige Hilfsmittel. Wir starten mit gegebenen endlichen Monoiden $M_1, M_2, \ldots, M_\ell$ und einem Spurmonoid $\mathbb{M}(\Sigma, I)$. Aus $\mathbb{M}$ erhalten wir das Skeletonmonoid $\mathbb{S}$ und den gleichnamigen starken I-Homomorphismus $\mathbb{S}$ von $\mathbb{M}$ ins Skeletonmonoid. Des weiteren sind Abbildungen $\mathbb{S}_1, \mathbb{S}_2, \ldots, \mathbb{S}_\ell$ gegeben. Für alle $1 \leq i \leq \ell$ ist $\mathbb{S}_i : M_i \to \mathbb{S}$ ein starker I-Homomorphismus. Die Monoide $M_1 \ldots, M_\ell$ werden dadurch zu I-Monoiden und Elemente aus $\mathbb{M}, M_1, \ldots, M_\ell$ können wie in Abschnitt 2.4 beschrieben paarweise auf Unabhängigkeit geprüft werden.
Wir werden im Folgenden Tupel aus $\Sigma \cup \{\varepsilon\} \times M_1 \times \ldots \times \Sigma \cup \{\varepsilon\} \times M_\ell$ betrachten. Dafür definieren wir die Unabhängigkeitsrelation I für Tupel $x = (b_1, x_1, \ldots, b_\ell, x_\ell), y = (c_1, y_1, \ldots, c_\ell, y_\ell) \in \Sigma \cup \{\varepsilon\} \times M_1 \times \ldots \times \Sigma \cup \{\varepsilon\} \times M_\ell$ als

$$(x, y) \in I \Leftrightarrow \forall\, 1 \leq i, j \leq \ell : (x_i, y_j), (x_i, c_j), (b_i, y_j), (b_i, c_j) \in I$$

Dies lässt sich wie folgt darstellen:





Die Tupel x und y sind also genau dann unabhängig, wenn alle Komponenten voneinander unabhängig sind. Wir sagen x und y sind in allen verschiedenen Komponenten unabhängig, wenn für alle $1 \leq i \neq j \leq \ell$ gilt $(x_i, y_j) \in I$ und $(b_i, c_j) \in I$, sowie für alle $1 \leq i, j \leq \ell$ gilt $(x_i, c_j) \in I$ und $(b_i, y_j) \in I$. Als nächstes erweitern wir die Skeleton Abbildung auf solche Tupel $x = (b_1, x_1, \ldots, b_\ell, x_\ell)$, indem wir $\mathbb{S}(x)$ definieren als

$$\mathbb{S}(x) := \mathbb{S}(b_1) \cdot \mathbb{S}_1(x_1) \cdot \ldots \cdot \mathbb{S}(b_\ell) \cdot \mathbb{S}_2(x_\ell)$$

Es lässt sich mit Hilfe der Definition von I-Monoiden leicht überprüfen, dass die so definierte Abbildung ein starker I-Homomorphismus ist.
Damit können wir zur Definition des Schützenberger-Produkts für Spursprachen kommen. Es sei ein Spurmonoid $\mathbb{M}(\Sigma, I)$ gegeben. Weiter seien endliche Monoide $M_1, \ldots, M_\ell$ und zugehörige starke I-Homomorphismen ins Skeletonmonoid von $\mathbb{M}(\Sigma, I)$ gegeben. Wir definieren das Schützenberger-Produkt $\diamond(\Sigma \cup \{\varepsilon\}, M_1, \ldots, \Sigma \cup \{\varepsilon\}, M_\ell)$ als Teilmenge von $2^{\Sigma \cup \{\varepsilon\} \times M_1 \times \ldots \times \Sigma \cup \{\varepsilon\} \times M_\ell}$. Ein Element $A \in \diamond$ ist also eine Menge von Tupeln der Form $(x_0, b_1, x_1, \ldots, b_\ell, x_\ell) \in \Sigma \cup \{\varepsilon\} \times M_1 \times \ldots \times \Sigma \cup \{\varepsilon\} \times M_\ell$. Eine Menge $A \subseteq \Sigma \cup \{\varepsilon\} \times M_1 \times \ldots \times \Sigma \cup \{\varepsilon\} \times M_\ell$ ist genau dann im Schützenberger-Produkt $\diamond$ enthalten, wenn gilt:

- Für alle $1 \leq i \leq \ell$ existiert in A genau ein Tupel der Form $(1, \ldots, 1, x_i, 1, \ldots, 1)$
- Für alle $x, y \in A$ gilt $\mathbb{S}(x) = \mathbb{S}(y)$
- Seien $x = (b_1, x_1, \ldots, b_\ell, x_\ell), y = (c_1, y_1, \ldots, c_\ell, y_\ell) \in A$ mit x und y sind unabhängig in allen verschiedenen Komponenten. Dann folgt $x_i = y_i$ und $b_i = c_i$ für alle $1 \leq i \leq \ell$.

Das Produkt $x \cdot y$ von Elementen $x = (b_1, x_1, \ldots, b_\ell, x_k), y = (c_1, y_1, \ldots, c_\ell, y_\ell) \in \Sigma \cup \{\varepsilon\} \times M_1 \times \ldots \times \Sigma \cup \{\varepsilon\} \times M_\ell$ ist definiert als

$$x \cdot y := (b_1 c_1, x_1 y_1, \ldots, b_\ell c_\ell, x_\ell y_\ell)$$

falls gilt

- Für alle $1 \leq i \leq \ell$ ist $b_i = 1$ oder $c_i = 1$
- Für alle $1 \leq j < i \leq \ell$ ist $(y_i, x_j) \in I$ und $(y_i, b_j) \in I$
- Für alle $1 \leq j \leq i \leq \ell$ ist $(c_i, x_j) \in I$ und $(c_i, b_j) \in I$

Andernfalls ist $x \cdot y$ undefiniert. Das Produkt $A \cdot B$ für $A, B \in \diamond$ ist definiert als

$$A \cdot B := \{\, x \cdot y \mid x \in A, y \in B, x \cdot y \text{ ist definiert} \,\}$$

Wir bezeichnen das so definierte Schützenberger-Produkt von I-Monoiden im Folgenden auch als erweiteres Schützenberger-Produkt. An dieser Stelle soll die Idee dieser Definition kurz erläutert werden. Das Grundproblem, das mit Hilfe des erweiterten Schützenberger-Produkts gelöst werden soll, ist für eine Spur s die Zugehörigkeit zu

einer Spursprache der Form $L = a_1L_1 \dots a_\ell L_\ell$ zu prüfen. Dafür muss getestet werden, ob eine Vertauschung der Zeichen in s existiert, so dass gilt $s = a_1t_1 \dots a_\ell t_\ell$ und $t_i \in L_i$ für alle $1 \leq i \leq \ell$. Das Schützenberger Produkt bildet ein Konstrukt, in dem alle möglichen Aufteilungen einer Spur s gespeichert werden können. In einem weiteren Schritt kann dann überprüft werden, ob darin eine passende Zerlegung für L enthalten ist. Die Definition des Produkts $x \cdot y$ berücksichtigt, dass Zeichen aus Komponente i von y an Zeichen in den Komponenten j von x für $1 \leq i < j \leq \ell$ vorbei getauscht werden müssen. Ist dies nicht möglich, so ist das Produkt undefiniert.
Es muss noch gezeigt werden, dass das Schützenberger-Produkt $\diamond$ mit der Verknüpfung $\cdot$ wohldefiniert ist. Aus $A, B \in \diamond$ muss folgen, dass $A \cdot B \in \diamond$ gilt. Für alle $1 \leq i \leq \ell$ existiert in A genau ein Tupel $x = (1, \dots, 1, x_i, 1, \dots, 1)$ und genau ein Tupel $y = (1, \dots, 1, y_i, 1, \dots, 1)$ in B. Es ist $x \cdot y = (1, \dots, 1, x_iy_i, 1, \dots, 1) \in A \cdot B$ und $x \cdot y$ ist das einzige Tupel dieser Form in $A \cdot B$. Ist $x \cdot y \in A \cdot B$ sowie $x' \cdot y' \in A \cdot B$, so folgt $\mathbb{S}(x \cdot y) = \mathbb{S}(x) \cdot \mathbb{S}(y) = \mathbb{S}(x') \cdot \mathbb{S}(y') = \mathbb{S}(x' \cdot y')$. Angenommen es seien $x \cdot y \in A \cdot B$ und $x' \cdot y' \in A \cdot B$ und $x \cdot y$ sei unabhängig zu allen verschiedenen Komponenten von $x' \cdot y'$. Da Abhängigkeiten von x und x' auch nach Heranmultiplizieren von y und y' erhalten bleiben, müssen bereits x und x' in allen verschiedenen Komponenten unabhängig sein. Da $x, x' \in A \in \diamond$ folgt daraus $x = x'$ und analog $y = y'$. Also ist $x \cdot y = x' \cdot y'$. Insgesamt ergibt dies $A \cdot B \in \diamond$.
Die Unabhängigkeitsrelation auf $\diamond$ ergibt sich aus der bereits definierten Relation für Tupel aus $\Sigma \cup \{\varepsilon\} \times M_1 \times \dots \times \Sigma \cup \{\varepsilon\} \times M_\ell$. Es ist $(A, B) \in I$ für $A, B \in \diamond$, falls für alle $x \in A$ und für alle $y \in B$ gilt $(x, y) \in I$. Es sei angemerkt, dass aus $(x, y) \in I$ für ein $x \in A$ und ein $y \in B$ bereits die Unabhängigkeit für alle (x', y') mit $x' \in A, y' \in B$ und damit für (A, B) folgt. Denn es ist $(\mathbb{S}(x), \mathbb{S}(y)) \in I$ und $\mathbb{S}(x) = \mathbb{S}(x')$ sowie $\mathbb{S}(y) = \mathbb{S}(y')$ für alle $x' \in A, y' \in B$.

3.2 Spursprachen des Schützenberger-Produkts

Das Schützenberger-Produkt von Monoiden erkennt genau Vereinigungen von markierten Produkten von Wortsprachen. Wir wollen nun näher untersuchen, welche Spursprachen vom Schützenberger-Produkt endlicher I-Monoide erkannt werden. Wir beweisen hierzu zunächst folgenden Satz:

Satz 3.1 *Seien $M_1, \dots, M_\ell$ endliche Monoide und (Σ, I) ein endliches Unabhängigkeitsalphabet. Das Schützenberger Produkt $\diamond(\Sigma \cup \{\varepsilon\}, M_1, \dots, \Sigma \cup \{\varepsilon\}, M_\ell)$ erkennt Sprachen der Form $L = a_1L_1 \dots a_\ell L_\ell$. Für alle $1 \leq i \leq \ell$ gilt dabei $a_i \in \Sigma \cup \{\varepsilon\}$ und L_i ist eine von M_i durch einen starken I-Homomorphismus erkannte Spursprache.*

Wir starten nun mit ℓ gegebenen Markern a_i, Spursprachen L_i, sowie starken I-Homomorphismen $\varphi_i : \mathbb{M}(\Sigma, I) \to M_i$, die L_i erkennen. Es gilt $\varphi_i^{-1}\varphi_i(L_i) = L_i$ für $1 \leq i \leq \ell$. Durch die starken I-Homomorphismen φ_i ergibt sich eine Unabhängigkeitsrelation auf M_i, die Monoide M_i werden I-Monoide. Es lassen sich, wie in Abschnitt 2.4 beschrieben, alle Elemente aus den Monoiden $\mathbb{M}, M_1, \dots, M_\ell$ paarweise auf Unabhängigkeit vergleichen. Unser Ziel ist die Definition einer Teilmenge $P \subseteq \diamond$, die $L = a_1L_1 \dots a_\ell L_\ell$

erkennt. Hierzu konstruieren wir aus den Homomorphismen φ_i einen Homomorphismus $\varphi : \mathbb{M} \to \diamond$ für Zeichen $b \in \Sigma$ wie folgt

$$\begin{aligned}\varphi(b) := &\{ (1, \ldots, 1, x_i, 1, \ldots, 1) \mid x_i = \varphi_i(b) \} \\ &\cup \{ (1, \ldots, 1, b, 1, \ldots, 1) \mid b = a_i \}\end{aligned}$$

Für eine Spur t ergibt sich

$$\varphi(t) = \{ (b_1, x_1, \ldots, b_\ell, x_\ell) \mid \exists b_1 t_1 \ldots b_\ell t_\ell = t \text{ mit } \forall i : x_i = \varphi_i(t_i), b_i = 1 \vee b_i = a_i \}$$

In $\varphi(t)$ werden also alle beliebigen Zerlegungen einer Spur t gespeichert (mit der Einschränkung, dass der Marker an Position i nur ε oder a_i sein kann). Die Spur t ist genau dann Element von $L = a_1 L_1 \ldots a_\ell L_\ell$, wenn es eine Zerlegung $a_1 t_1 \ldots a_\ell t_\ell$ von t mit $t_i \in L_i$ für $1 \leq i \leq \ell$, bzw. $\varphi_i(t_i) \in \varphi_i(L_i)$, gibt. Setze daher

$$\begin{aligned}X_i &:= \varphi_i(L_i) \\ X &:= a_1 \times X_1 \times \ldots \times a_\ell \times X_\ell \\ P &:= \{ A \in \diamond \mid A \cap X \neq \emptyset \}\end{aligned}$$

Es gilt $\varphi^{-1}(P) = L$, die Teilmenge $P \subseteq \diamond$ erkennt genau die Sprache L:

$$\begin{aligned}\varphi^{-1}(P) &= \{ t \in \mathbb{M} \mid \varphi(t) \in P \} \\ &= \{ t \in \mathbb{M} \mid \varphi(t) \cap X \neq \emptyset \} \\ &= \{ t \in \mathbb{M} \mid \exists b_1 t_1 \ldots b_\ell t_\ell = t \text{ mit } \forall 1 \leq i \leq \ell : \varphi_i(t_i) \in X_i, b_i = a_i \} \\ &= \{ t \in M \mid \exists a_1 t_1 \ldots a_\ell t_\ell = t \text{ mit } \forall 1 \leq i \leq \ell : t_i \in L_i \} \\ &= L\end{aligned}$$

Es muss noch gezeigt werden, dass φ starker I-Homomorphismus ist und von $\mathbb{M}$ nach $\diamond$ abbildet.

Lemma 3.2 (φ Homomorphismus) *Die Abbildung φ ist ein Homomorphismus, für alle Spuren $s, t \in \mathbb{M}$ gilt $\varphi(st) = \varphi(s) \cdot \varphi(t)$.*

Beweis: Nach Definition von φ ist

$$\varphi(st) = \{ (b_1, x_1, \ldots, b_\ell, x_\ell) \mid \exists b_1 u_1 \ldots b_k u_k = st \text{ mit } x_i = \varphi_i(u_i), b_i = \varepsilon \vee b_i = a_i \}$$

Es gilt

$$\begin{aligned}\varphi(s) \cdot \varphi(t) = &\{ (c_1, y_1, \ldots, c_\ell, y_\ell) \mid \exists c_1 \ldots c_\ell s_\ell = s \text{ mit } y_i = \varphi_i(s_i), c_i = \varepsilon \vee c_i = a_i \} \\ &\cdot \{ (d_1, z_1, \ldots, d_\ell, z_\ell) \mid \exists d_1 t_1 \ldots d_\ell t_\ell = t \text{ mit } z_i = \varphi_i(t_i), d_i = \varepsilon \vee d_i = a_i \} \\ = &\{ (c_1 d_1, y_1 z_1, \ldots, c_\ell d_\ell, y_\ell, z_\ell) \}\end{aligned}$$

Dabei muss für die letzte Zeile gelten, dass $c_1 s_1 \ldots c_\ell s_\ell = s$ und $d_1 t_1 \ldots d_\ell t_\ell = t$ existieren, mit $\varphi_i(s_i) = y_i$ sowie $\varphi_i(t_i) = z_i$. Weiter ist $c_i d_i = \varepsilon \vee c_i d_i = a_i$. Außerdem gelten

noch die Unabhängigkeitsbedingungen $\forall \ell \geq j > i \geq 1 : (t_i, s_j) \in I, (t_i, c_j) \in I$ und $\forall \ell \geq j \geq i \geq 1 : (d_i, s_j) \in I, (d_i, c_j) \in I$.
Das sogenannte *Levi Lemma* [5] besagt:

$$\begin{aligned} b_1 u_1 \dots b_\ell u_\ell = st \Leftrightarrow & \exists c_1 s_1 \dots c_\ell s_\ell = s,\ \exists d_1 t_1 \dots d_\ell t_\ell = t \\ & u_i = s_i t_i, b_i = c_i d_i \\ & \forall \ell \geq j > i \geq 1 : (s_j, t_i), (t_j, c_i) \in I \\ & \forall \ell \geq j > i \geq 1 : (c_j, d_i) \in I, \forall j \geq i (s_j, d_i) \in I \end{aligned}$$

Für den Beweis dieses Lemmas sei auf [5] verwiesen. Schränken wir die linke Seite durch $b_i = \varepsilon \vee b_i = a_i$ ein, so ergibt sich mit $b_i = c_i d_i$ auf der rechten Seite $c_i = \varepsilon \vee d_i = \varepsilon$. Damit ist $(c_i, d_i) \in I$ für alle i und es folgt die Behauptung. □

Lemma 3.3 (φ starker I-Homomorphismus) *φ ist starker I-Homomorphismus, es gilt*

$$(s, t) \in I \Leftrightarrow (\varphi(s), \varphi(t)) \in I$$

Beweis: Spuren s und t sind nach Definition genau dann unabhängig, falls alle Zeichen von s und t wechselseitig unabhängig sind, also falls $alph(s) \times alph(t) \subseteq I$. Für $(\varphi(s), \varphi(t))$ gilt nach Definition

$$\begin{aligned} (\varphi(s), \varphi(t)) \in I \Leftrightarrow & \forall (b_1, x_1, \dots, b_\ell, x_\ell) \in \varphi(s), \forall (c_1, y_1, \dots, c_\ell, y_\ell) \in \varphi(t) \\ & \forall 1 \leq i, j \leq \ell : (x_i, y_j), (x_i, c_j), (b_i, y_j), (b_i, c_j) \in I \end{aligned}$$

Nach Definition von φ und da alle Abbildungen φ_i starke I-Homomorphismen sind, ist dies äquivalent zu

$$\begin{aligned} (\varphi(s), \varphi(t)) \in I \Leftrightarrow & \forall b_1 s_1 \dots b_\ell s_\ell = s \text{ mit } \varphi_i(s_i) = x_i \\ & \forall c_1 t_1 \dots c_\ell t_\ell = t \text{ mit } \varphi_i(t_i) = y_i \text{ gilt} \\ & \forall i, j : (\varphi_i(s_i), \varphi_j(t_j)), (\varphi_i(s_i), c_j), (b_i, \varphi_j(t_j)), (b_i, c_j) \in I \\ \Leftrightarrow & \forall b_1 s_1 \dots b_\ell s_\ell = s, \forall c_1 t_1 \dots c_\ell t_\ell = t \text{ gilt} \\ & \forall i, j : (s_i, t_j), (s_i, c_j), (b_i, t_j), (b_i, c_j) \in I \\ \Leftrightarrow & (s, t) \in I \end{aligned}$$

Für die Indizees i, j gilt $1 \leq i, j \leq \ell$. □

Lemma 3.4 (φ bildet nach $\diamond$ ab) *Der Homomorphismus φ bildet nach $\diamond$ ab*

Beweis: Sei $a \in \Sigma$ und $A = \varphi(a)$. Jede Menge $B \in \diamond$ muss nach Definition des Schützenberger-Produkts für alle i genau ein Tupel der Form $(1, \ldots, 1, x_i, 1, \ldots, 1)$ enthalten. Aus der Definition von φ folgt direkt, dass genau ein Tupel dieser Form in A existiert, nämlich $(1, \ldots, 1, \varphi_i(a), 1, \ldots, 1)$. Weiter ist zu zeigen, dass für alle $x, y \in A$ gilt $\mathbb{S}(x) = \mathbb{S}(y)$. Dies folgt aus der Definition der Skeleton Abbildungen, denn danach gilt $\mathbb{S}(a) = \mathbb{S}(\varphi_i(a)) = \mathbb{S}(\varphi_j(a))$. Seien $x, y \in \varphi(a)$ mit x und y sind in allen verschiedenen Komponenten unabhängig. Es muss gezeigt werden, dass daraus $x = y$ folgt. Die Tupel x und y können jeweils die Form $(1, \ldots, 1, x_i, 1, \ldots, 1)$ oder $(1, \ldots, 1, a_i, 1, \ldots, 1)$ haben. Haben x und y die Form $(1, \ldots, 1, x_i, 1, \ldots, 1)$ und den Eintrag an der selben Stelle i, so haben wir die Eindeutigkeit bereits gezeigt. Dies folgt analog, falls x und y die Form $(1, \ldots, 1, a_i, 1, \ldots, 1)$ mit Eintrag an der gleichen Position haben. Falls x und y Einträge an unterschiedlichen Positionen haben, so folgt aus $\mathbb{S}(x) = \mathbb{S}(y)$, dass diese Einträge abhängig voneinander sind. Dies ist ein Widerspruch zur Voraussetzung. Insgesamt folgt $A \in \diamond$. □

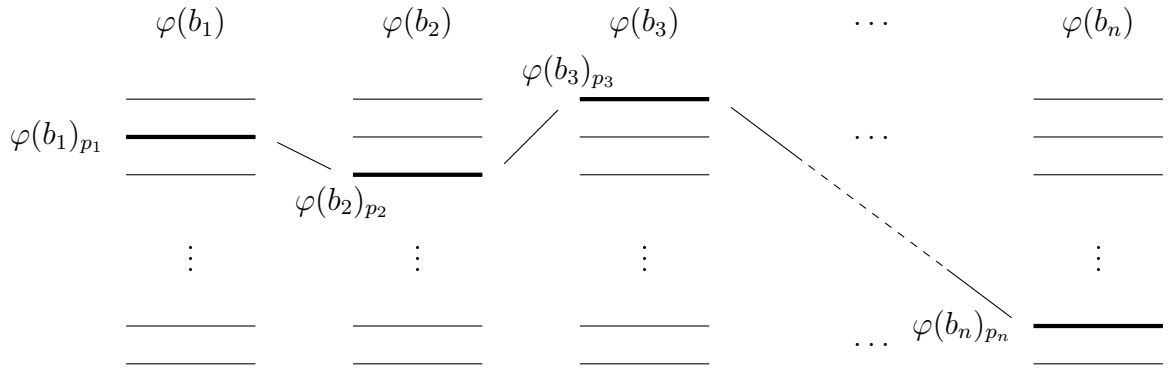
Satz 3.1 besagt, dass das Schützenberger-Produkt von I-Monoiden das markierte Produkt von Spursprachen erkennt. Im nächsten Schritt wird gezeigt, dass sich jede vom Schützenberger-Produkt erkannte Spursprache als Vereinigung solcher markierten Produkte darstellen lässt.

Satz 3.5 *Gegeben sei das Spurmonoid $\mathbb{M}(\Sigma, I)$ zu einem gegebenen Unabhängigkeitsalphabet (Σ, I) und endliche Monoide $M_1, \ldots, M_\ell$ mit starken I-Homomorphismen $\mathbb{S}_i : M_i \to \mathbb{S}(\Sigma, I)$. Sei L eine von $\diamond(\Sigma \cup \{\varepsilon\}, M_1, \ldots, \Sigma \cup \{\varepsilon\}, M_\ell)$ erkannte Sprache, also $\varphi^{-1}\varphi(L) = L$ für einen starken I-Homomorphismus $\varphi : \mathbb{M} \to \diamond$. Die Sprache L lässt sich dann darstellen als endliche boolsche Kombination von Sprachen der Form*

$$a_1 L_1 \ldots a_k L_k \tag{3.5.1}$$

mit $0 \leq k \leq |\Sigma \cup \{\varepsilon\}| + |M_1| + |\Sigma \cup \{\varepsilon\}| + |M_2| + \ldots + |M_{\ell-1}| + |\Sigma \cup \{\varepsilon\}|$ und $a_i \in \Sigma$ für alle $1 \leq i \leq k$. Des weiteren ist L_i für alle $1 \leq i \leq k$ eine von $M_1 \times M_2 \times \ldots \times M_\ell$ erkannte Sprache.

Mit Hilfe der Skeleton Abbildungen können wir im Folgenden Elemente aus den Monoiden $\mathbb{M}, M_1, \ldots, M_\ell$ auf Unabhängigkeit prüfen. Für $\varphi(t)$ gilt offensichtlich $|\varphi(t)| \leq |\Sigma \cup \{\varepsilon\}| \cdot |M_1| \cdot \ldots \cdot |\Sigma \cup \{\varepsilon\}| \cdot |M_\ell|$, da es nur $|\Sigma \cup \{\varepsilon\}| \cdot |M_1| \cdot \ldots \cdot |\Sigma \cup \{\varepsilon\}| \cdot |M_\ell|$ viele verschiedene Tupel gibt. Wir können für alle Spuren t auf $\varphi(t)$ eine lineare Ordnung definieren und so die Elemente von $\varphi(t)$ durchnummerieren. $\varphi(t)_i \in \Sigma \cup \{\varepsilon\} \times M_1 \times \ldots \times \Sigma \cup \{\varepsilon\} \times M_\ell$ ist dann das i-te Element in $\varphi(t)$ mit $1 \leq i \leq |\varphi(t)|$. Da φ ein Homomorphismus ist, gilt $\varphi(t) = \varphi(t_1) \ldots \varphi(t_j)$ für $t = t_1 \ldots t_j$. Im Speziellen ist $\varphi(t) = \varphi(b_1) \ldots \varphi(b_n)$ für $t = b_1 \ldots b_n$. Ein Element $\varphi(t)_i$ ergibt sich als Produkt von Elementen $\varphi(b_1)_{p_1}, \varphi(b_2)_{p_2}, \ldots, \varphi(b_n)_{p_n}$, wobei $1 \leq p_i \leq |\varphi(b_i)|$ gilt.



Diese Darstellung motiviert die Definition von *Pfaden*. Unter einem Pfad p über einer Spur $t = t_1 t_2 \ldots t_j$ verstehen wir ein j-Tupel mit Einträgen p_i aus $\{1, \ldots, m\}$, $m := |\Sigma \cup \{\varepsilon\}| \cdot |M_1| \cdot \ldots \cdot |\Sigma \cup \{\varepsilon\}| \cdot |M_\ell|$. Danach ergibt sich $\varphi(t) = \varphi(b_1) \cdot \ldots \cdot \varphi(b_n)$ für $t = b_1 \ldots b_n$ als Vereinigung über allen Pfaden p

$$\varphi(t) = \bigcup_p \varphi(b_1)_{p_1} \cdot \varphi(b_2)_{p_2} \cdot \ldots \cdot \varphi(b_n)_{p_n} \tag{3.5.2}$$

Hier kann es Pfade geben, deren Produkt auf Grund von Abhängigkeiten undefiniert ist. Außerdem können Indizes p_i mit $p_i > |\varphi(b_i)|$ auftreten. In beiden Fällen trägt der Pfad nichts zur Vereinigung bei. Wir interessieren uns im Folgenden nur für Pfade, deren Produkt definiert ist. Unser nächstes Ziel ist es, mit Hilfe dieser Darstellung jeden Pfad p von t, der zu einem definierten Produkt führt, in endlich viele Blöcke zu zerlegen. Bei den Tupeln innerhalb eines Blocks sollen keine Abhängigkeiten zwischen unterschiedlichen Komponenten auftreten und alle Marker Einträge der im Block auftretenden Tupel sollen 1 sein. Das Produkt eines Blocks soll sich also verhalten wie ein Element aus dem direkten Produkt $M_1 \times M_2 \times \ldots \times M_\ell$. Die Blöcke werden von Zeichen $a_j \in \Sigma$ getrennt. Diese Trennzeichen a_j bezeichnen wir als *Marker*, das zugehörige Tupel $\varphi(a_j)_{q_j}$ als *Markertupel*. Um $\varphi(t)_p$ entsprechend in Blöcke aufteilen zu können, brauchen wir noch einige Hilfsmittel.
Seien $x = (x_0, b_1, x_1, \ldots, b_\ell, x_\ell), y \in \Sigma \cup \{\varepsilon\} \times M_1 \times \ldots \times \Sigma \cup \{\varepsilon\} \times M_\ell$ mit $x \cdot y$ definiert. Damit sind die möglichen Einträge in jeder Komponente von y beschränkt. Wir wollen zu x die Mengen an Elementen bestimmen, die in den Komponenten von y auftreten können, so dass nicht auf Grund von Abhängigkeiten das Produkt undefiniert wird. Wir definieren dazu das *Potential* λ von x als

$$\lambda(x) = (B_1, X_1, \ldots, B_{\ell-1}, X_{\ell-1}, B_\ell)$$

mit $X_i \subseteq M_i$ für $1 \leq i \leq \ell - 1$ und $B_j \subseteq \Sigma \cup \{\varepsilon\}$ für $1 \leq j \leq \ell$, sowie

$$\begin{aligned} a \in X_i &\Leftrightarrow \forall 1 \leq i < j \leq \ell : (a, x_j) \in I \text{ und } (a, b_j) \in I \\ a \in B_i &\Leftrightarrow a = 1 \text{ falls } b_i \neq 1 \text{ und } \forall 1 \leq i \leq j \leq \ell : (a, x_j), (a, b_j) \in I \end{aligned}$$

Wir wollen auf λ noch eine Halbordnung „$\leq$ “ definieren. Seien also $\lambda(x) = (B_1, X_1, \ldots, B_{\ell-1}, X_{\ell-1}, B_\ell)$ und $\lambda(y) = (C_1, Y_1, \ldots, C_{\ell-1}, Y_{\ell-1}, C_\ell)$. Es ist $\lambda(y) \leq \lambda(x)$ falls $Y_i \subseteq X_i$ sowie $C_j \subseteq B_j$.

Lemma 3.6 *Seien $x, y \in \Sigma \cup \{\varepsilon\} \times M_1 \times \ldots \times \Sigma \cup \{\varepsilon\} \times M_\ell$ mit $x \cdot y$ definiert. Dann gilt*

(a) $\lambda(xy) \leq \lambda(x)$

(b) $\lambda(xy) = \lambda(x) \Rightarrow$ *Es ist* $y = (1, y_1, 1, y_2, \ldots, 1, y_\ell)$ *mit voneinander unabhängigen Komponenten* y_i

Beweis: Im Folgenden sei $x = (b_1, x_1, \ldots, b_\ell, x_\ell)$ und $y = (c_1, y_1, \ldots, c_\ell, y_\ell)$ mit $x \cdot y$ definiert. Die Potentiale von x und xy seien $\lambda(x) = (B_1, X_1, \ldots, B_{\ell-1}, X_{\ell-1}, B_\ell)$ und $\lambda(xy) = (C_1, Y_1, \ldots, C_{\ell-1}, Y_{\ell-1}, C_\ell)$.
(a): Es ist $x \cdot y = (b_1c_1, x_1y_1, \ldots, b_\ell c_\ell, x_\ell y_\ell)$. Es gilt $a \in Y_i$ genau dann, wenn a von allen x_jy_j und b_jc_j unabhängig ist für alle $j > i$. Daraus folgt, dass a auch von allen x_j und b_j unabhängig ist für alle $1 \leq i < j \leq \ell$. Damit gilt $a \in X_i$ und $Y_i \subseteq X_i$. Falls $b_ic_i \neq 1$, so ist $C_i = \{1\}$ und $C_i \subseteq B_i$. Sei nun $b_ic_i = 1$, also auch $b_i = 1$. Es ist $a \in C_i$ genau dann, wenn a von allen x_jy_j für $1 \leq i \leq j \leq \ell$ sowie von allen b_jc_j für $1 \leq i < j \leq \ell$ unabhängig ist. Daraus folgt, dass a von allen x_j für $j \geq i$ sowie von allen b_j für $j > i$ unabhängig ist. Also ist $a \in B_i$ und auch in diesem Fall gilt $C_i \subseteq B_i$. Insgesamt ergibt sich $\lambda(xy) \leq \lambda(x)$.
(b): Wir zeigen zunächst, dass gilt $c_i = 1$ für alle $1 \leq i \leq \ell$. Angenommen es existiert ein $c_i \neq 1$ in y. Da $x \cdot y$ definiert ist, gilt $b_i = 1$ und c_i ist unabhängig von allen x_j für $j \geq i$, sowie von allen b_j für $j > i$ (da $b_i = 1$ gilt die Aussage sogar für $j \geq i$).

$$(b_1,\ x_1,\ \ldots,\ 1,\ x_i,\ \ldots,\ b_j,\ x_j,\ \ldots,\ b_\ell,\ x_\ell)$$

$$(c_1,\ y_1,\ \ldots,\ c_i,\ y_i,\ \ldots,\ c_j,\ y_j,\ \ldots,\ c_\ell,\ y_\ell)$$

Daraus folgt, dass c_i in der Komponente B_i des Potentials $\lambda(x)$ liegt. Allerdings wegen $b_ic_i = c_i \neq 1$ gilt $c_i \notin C_i = \{1\}$. Es ist $B_i \neq C_i$ im Widerspruch zur Annahme $\lambda(x) = \lambda(xy)$. Also hat y die Form $(1, y_1, 1, y_2 \ldots, 1, y_\ell)$. Es bleibt zu zeigen, dass für alle $1 \leq i \neq j \leq \ell$ gilt y_i und y_j sind unabhängig. Sei $i < j$ mit $(y_i, y_j) \notin I$. Weil $x \cdot y$ definiert ist, gilt y_i ist unabhängig zu allen x_k und allen b_k für $1 \leq i < k \leq \ell$.

$$(b_1,\ x_1,\ \ldots,\ b_i,\ x_i,\ \ldots,\ b_j,\ x_j,\ \ldots,\ b_\ell,\ x_\ell)$$

$$(c_1,\ y_1,\ \ldots,\ c_i,\ y_i,\ \ldots,\ c_j,\ y_j,\ \ldots,\ c_\ell,\ y_\ell)$$

Daher ist y_i in der Komponente B_i des Potentials $\lambda(x)$. Aus $(y_i, y_j) \notin I$ folgt $(y_i, x_jy_j) \notin I$. Also ist y_i nicht in der Komponente C_i des Potentials $\lambda(xy)$, im Widerspruch zur Annahme $\lambda(x) = \lambda(xy)$. $\square$

Wir zerlegen nun ein definiertes Produkt $\varphi(b_1)_{p_1} \cdot \ldots \cdot \varphi(b_n)_{p_n}$ aus $\varphi(t)$ in Blöcke und Markertupel. Wir durchlaufen das Produkt von links nach rechts und betrachten immer größere Stücke $\varphi(b_1)_{p_1} \cdot \ldots \cdot \varphi(b_i)_{p_i}$. Sei $x = \varphi(b_1)_{p_1} \cdot \ldots \varphi(b_i)_{p_i}$. Nach Lemma 3.6 gilt $\lambda(x \cdot \varphi(b_{i+1})_{p_{i+1}}) \leq \lambda(x)$. Jedes Mal, wenn das Potential durch heranmultiplizieren eines weiteren Tupels $\varphi(b_{i+1})_{p_{i+1}}$ abnimmt, wollen wir dieses Tupel zum Markertupel machen. Das erste Tupel $\varphi(b_1)_{p_1}$ machen wir dabei kanonisch zum Markertupel. Da das Potential höchstens $|\Sigma \cup \{\varepsilon\}| + |M_1| + |\Sigma \cup \{\varepsilon\}| + |M_2| + \ldots + |M_{\ell-1}| + |\Sigma \cup \{\varepsilon\}|$ mal abnehmen kann, werden nur endlich viele Tupel zu Markern. Nimmt das Potential durch heranmultiplizieren von $y = \varphi(b_{i+1})_{p_{i+1}} \cdot \varphi(b_j)_{p_j}$ nicht ab, so hat nach Lemma 3.6 das Tupel y die Form $(1, y_1, 1, y_2, \ldots, 1, y_\ell)$ mit $(y_i, y_j) \in I$ für alle $1 \leq i \neq j \leq \ell$. Zusammen mit (3.5.2) ergibt sich daraus folgende Darstellung für $\varphi(t)$

$$\varphi(t) = \bigcup_p \varphi(a_1)_{q_1} \varphi(t_1)_{p_1} \ldots \varphi(a_k)_{q_k} \varphi(t_k)_{p_k} \tag{3.6.1}$$

mit $1 \leq k \leq |\Sigma \cup \{\varepsilon\}| + |M_1| + |\Sigma \cup \{\varepsilon\}| + |M_2| + \ldots + |M_{\ell-1}| + |\Sigma \cup \{\varepsilon\}|$ und $p_i, q_i \in \{1, \ldots, m\}$ für $m := |\Sigma \cup \{\varepsilon\}| \cdot |M_1| \cdot \ldots \cdot |\Sigma \cup \{\varepsilon\}| \cdot |M_\ell|$. Weiter muss gelten $t = a_1 t_1 \ldots a_k t_k$, sowie $\varphi(t_i)_{p_i}$ ist ein Tupel, in dem alle Komponenten voneinander unabhängig sind und alle Einträge aus $\Sigma \cup \{\varepsilon\}$ gleich 1 sind. Diese Darstellung bringt uns zur Definition von *Signaturen*. Eine Signatur σ ist ein Tupel

$$\sigma = (a_1, x_1, \ldots, a_k, x_k)$$

für $0 \leq k \leq |\Sigma \cup \{\varepsilon\}| + |M_1| + |\Sigma \cup \{\varepsilon\}| + |M_2| + \ldots + |M_{\ell-1}| + |\Sigma \cup \{\varepsilon\}|$. Dabei ist $x_i \in \times 1 \times M_1 \times 1 \times \ldots \times 1 \times M_\ell$ für alle $1 \leq i \leq k$ ein Tupel, dessen Komponenten alle unabhängig voneinander sind. Für $x_i = (1, y_1, 1, \ldots, 1, y_\ell)$ gilt also $(y_i, y_j) \in I$ für alle $i \neq j$. Für alle $1 \leq i \leq k$ ist a_i ein Zeichen aus Σ. Die Menge aller Signaturen bezeichnen wir mit $\mathscr{S}$. Da sowohl die Länge k einer Signatur, wie auch die Möglichkeiten für Einträge x_i und a_i endlich begrenzt sind, ist die Menge $\mathscr{S}$ endlich. Als *Inhalt* einer Spur t bezeichnen wir folgende Menge von Signaturen

$$cont(t) := \{\, \sigma = (a_1, x_1, \ldots, a_\ell, x_\ell) \in \mathscr{S} \;\mid\; \exists a_1 t_1 \ldots a_k t_k = t, \exists p : \varphi(t_i)_{p_i} = x_i \,\}$$

Als nächstes definieren wir hiermit eine Äquivalenzrelation auf dem Spurmonoid $\mathbb{M}$. Für $s, t \in \mathbb{M}$ sei

$$s \equiv t \text{ genau dann, wenn } cont(s) = cont(t)$$

Es lässt sich leicht prüfen, dass die so definierte Relation eine Äquivalenzrelation ist.

Lemma 3.7 $cont(s) = cont(t) \Rightarrow \varphi(s) = \varphi(t)$

Beweis: Wir zeigen zunächst $\varphi(s) \subseteq \varphi(t)$. Nach Gleichung (3.6.1) lässt sich $\varphi(s)$ darstellen als

$$\varphi(s) = \bigcup_p \varphi(a_1)_{q_1}\varphi(s_1)_{p_1} \dots \varphi(a_k)_{q_k}\varphi(s_k)_{p_k}$$

Wir zeigen nun, dass alle Tupel aus dieser Vereinigung auch in $\varphi(t)$ liegen. Sei also $\varphi(a_1)_{q_1}\varphi(s_1)_{p_1} \dots \varphi(a_k)_{q_k}\varphi(s_k)_{p_k}$ ein Tupel aus $\varphi(s)$. Wir definieren eine Signatur σ als

$$\sigma = (a_1, x_1, \dots, a_k, x_k)$$

mit $x_i = \varphi(s_i)_{p_i}$. Die Signatur σ liegt in $cont(s)$. Da $cont(s) = cont(t)$, gilt auch $\sigma \in cont(t)$. Daraus folgt, dass es eine Zerlegung $a_1t_1 \dots a_kt_k = t$ und einen Pfad p' gibt, mit $\varphi(t_i)_{p'_i} = x_i$. Die Einträge q'_i im Pfad p' setzen wir gleich q_i. Im Tupel $x_i = \varphi(t_i)_{p'_i}$ sind alle Komponenten voneinander unabhängig. Damit muss $\varphi(a_1)_{q'_1}\varphi(t_1)_{p'_1} \dots \varphi(a_k)_{q'_k} \in \varphi(t)$ gelten. Insgesamt ist

$$\varphi(a_1)_{q_1}\varphi(s_1)_{p_1} \dots \varphi(a_k)_{q_k} = \varphi(a_1)_{q'_1}\varphi(t_1)_{p'_1} \dots \varphi(a_k)_{q'_k} \in \varphi(t)$$

und es gilt $\varphi(s) \subseteq \varphi(t)$. Wegen Symmetrie folgt die Behauptung $\varphi(s) = \varphi(t)$. □

Sei $\pi : \mathbb{M} \to \mathbb{M} / \equiv$ die Projektion der Spuren auf deren Äquivalenzklassen. Da es nur endlich viele Objekte gibt, kann es auch nur endlich viele Äquivalenzklassen geben.

Korollar 3.8 *$cont(s) = cont(t) \Rightarrow \varphi^{-1}\varphi(s) = \varphi^{-1}\varphi(t)$. $\pi^{-1}\pi$ ist Verfeinerung von $\varphi^{-1}\varphi$, es gilt $\pi^{-1}\pi \subseteq \varphi^{-1}\varphi$.*

Jede von $\diamond$ erkannte Sprache L ist endliche Vereinigung von Klassen $\varphi^{-1}\varphi$. Eine Klasse $\varphi^{-1}\varphi$ ist wiederum endliche Vereinigung von Klassen $\pi^{-1}\pi$. Wir wollen zeigen, dass L die Form (3.5.1) hat. Dafür genügt es jetzt zu zeigen, dass Klassen $\pi^{-1}\pi$ diese Form haben. Wir definieren hierfür die *Sprache* einer Signatur σ als

$$L(\sigma) := \{\, t \in \mathbb{M} \mid \sigma \in cont(t) \,\}$$

Sprachen $L(\sigma)$ für ein $\sigma = (a_1, x_1, \dots, a_k, x_k)$ haben die Form (3.5.1), denn es gilt

$$\begin{aligned} L(\sigma) &= \{\, t \in \mathbb{M} \mid \sigma \in cont(t) \,\} \\ &= \{\, t \in \mathbb{M} \mid \exists a_1t_1 \dots a_kt_k = t, \exists p : \varphi(t_i)_{p_i} = x_i \,\} \\ &= a_1 \{\, t_1 \in \mathbb{M} \mid \exists p_1 : \varphi(t_1)_{p_1} = x_1 \,\} \\ &\quad \dots \\ &\quad a_k \{\, t_k \in \mathbb{M} \mid \exists p_k : \varphi(t_k)_{p_k} = x_k \,\} \\ &= a_1L_1 \dots a_kL_k \end{aligned}$$

und für alle $1 \leq i \leq k$ ist $x_i = (1, y_1, 1, y_2, \dots, 1, y_\ell)$ mit voneinander unabhängigen Einträgen y_j. Die Sprachen L_i sind definiert als

$$L_i = \{\, t_i \in \mathbb{M} \mid \exists p_i : \varphi(t_i)_{p_i} = x_i \,\} \tag{3.8.1}$$

Wir zeigen nun, dass die Sprachen L_i von $M_1 \times M_2 \times \dots \times M_\ell$ erkannt werden. Dazu brauchen wir ein weiteres Lemma.

Lemma 3.9 *Sei a ein beliebiges Zeichen aus Σ und $x = (1, x_1, 1, x_2, \ldots, 1, x_\ell)$ ein Tupel aus $M_1 \times 1 \times M_2 \times 1 \times \ldots \times 1 \times M_\ell$, bei dem alle Komponenten x_i voneinander unabhängig sind. Dann existiert höchstens ein Tupel $y \in \varphi(a)$ so, dass Elemente $z_1, z_2 \in 1 \times M_1 \times 1 \times M_2 \times \ldots \times 1 \times M_\ell$ existieren mit $z_1 y z_2 = x$.*

Beweis: Seien $y, y' \in \varphi(a)$ und $z_1, z_2, z_3, z_4 \in 1 \times M_1 \times 1 \times M_2 \times \ldots \times 1 \times M_\ell$ mit $z_1 y z_2 = x = z_3 y' z_4$. Da in x alle Komponenten voneinander unabhängig sind, sind auch alle Komponenten aus $z_1 y z_2$ zu verschiedenen Komponenten aus $z_3 y' z_4$ unabhängig. Würde es in y eine Komponente geben, die zu einer verschiedenen Komponente in y' abhängig wäre, so würde diese Abhängigkeit auch nach Heranmulitplizieren der Elemente z_1, z_2, z_3, z_4 bestehen. Daher sind y und y' in allen verschiedenen Komponenten voneinander unabhängig. Nach Definition des Schützenberger-Produkts folgt daraus $y = y'$. □

Korollar 3.10 *Angenommen es existiert ein Tupel $y \in \varphi(a)$ mit $z_1 y z_2 = x$ wie in Lemma 3.9 beschrieben. Dann ist y das einzige Tupel aus $\varphi(a)$, bei dem alle verschiedenen Komponenten in x und y voneinander unabhängig sind.*

Beweis: Wir zeigen zunächst, dass alle verschiedenen Komponenten in x und y voneinander unabhängig sind. Angenommen es gibt in y eine Komponente y_i, die zu einer Komponente x_j für $1 \leq i \neq j \leq \ell$ abhängig ist. Dann wäre die i-te Komponente aus $z_1 y z_2$ ebenfalls von x_j abhängig. Diese i-te Komponente entspricht wegen $z_1 y z_2 = x$ aber x_i, womit x_i und x_j abhängig wären. Dies ist ein Widerspruch. Also sind alle verschiedenen Komponenten aus x und y voneinander unabhängig. Angenommen es existiert ein weiteres Tupel $y' \neq y$ aus $\varphi(a)$, bei dem alle Komponenten von verschiedenen Komponenten von x unabhängig sind. Es muss verschiedene Komponenten in y und y' geben, die abhängig sind. Anderfalls müsste nach Definition des Schützenberger-Produkts $y = y'$ gelten. Da Abhängigkeiten beim Heranmultiplizieren von Elementen erhalten bleiben, ist eine Komponente von y' zu einer verschiedenen Komponente aus $z_1 y z_2 = x$ abhängig, im Widerspruch zur Annahme. □

Wir wollen nun zu gegebenem $x = (1, x_1, \ldots, 1, x_\ell)$ mit $(x_i, x_j) \in I$ für alle $i \neq j$, das Tupel $y \in \varphi(a)$ aus Lemma 3.9 bestimmen. Nach Korollar 3.10 müssen wir hierzu in $\varphi(a)$ nur nach einem Tupel suchen, bei dem alle Komponenten zu verschiedenen Komponenten aus x unabhängig sind. Wir wollen jetzt einen Homomorphismus $\psi : \mathbb{M} \to M_1 \times 1 \times M_2 \times 1 \times \ldots \times 1 \times M_\ell = M_1 \times M_2 \times \ldots \times M_\ell$ definieren, der eine Sprache $L = \{\, t \in \mathbb{M} \mid \exists p : \varphi(t)_p = x \,\}$ aus (3.8.1) erkennt. Dazu gehen wir wie folgt vor: Für alle Zeichen $a \in \Sigma$, suche in $\varphi(a)$ ein Tupel y, bei dem alle Komponenten zu verschiedenen Komponenten aus x unabhängig sind. Falls wir so ein Tupel y finden, setze $\psi(a) := y$. Andernfalls setze $\psi(a)$ auf ein beliebiges Tupel aus $\varphi(a)$. Es ist $t = b_1 b_2 \ldots b_n \in L$ genau dann, wenn $p_1, p_2, \ldots, p_n \in \{\, 1, 2, \ldots, m \,\}$ existieren, so dass gilt $\varphi(b_1)_{p_1} \cdot \varphi(b_2)_{p_2} \cdot \ldots \cdot \varphi(b_n)_{p_n} = x$. Nach Definition von ψ gilt $\psi(b_i) = \varphi(b_i)_{p_i}$. Also ist $t \in \psi^{-1}(x)$. Ist $t = b_1 b_2 \ldots b_n \in \psi^{-1}(x)$, dann existieren $p_1, p_2, \ldots, p_n$ mit $\varphi(b_i)_{p_i} = \psi(b_i)$

und t liegt in L. Damit erkennt ψ genau L, es gilt $\psi^{-1}(x) = L$. Demnach lassen sich alle Sprachen L_i aus (3.8.1) von $M_1 \times M_2 \times \ldots \times \mathbb{M}_\ell$ erkennen. Sprachen $L(\sigma)$ haben die gewünschte Form (3.5.1).

Lemma 3.11 *Für jede Spur t lässt sich $\pi^{-1}\pi(t)$ darstellen als*

$$\pi^{-1}\pi(t) = \bigcap_{\sigma \in cont(t)} L(\sigma) - \bigcup_{\sigma \in \mathscr{I} - cont(t)} L(\sigma)$$

Beweis: Aus den Definitionen folgt direkt, dass $\sigma \in cont(t)$ genau dann gilt, wenn t in $L(\sigma)$ liegt. Daher ist t im Schnitt enthalten. In $\pi^{-1}\pi(t)$ liegen alle Spuren s mit $cont(s) = cont(t)$. Diese Spuren s sind ebenfalls im Schnitt enthalten, da für alle $\sigma \in cont(t) = cont(s)$ folgt, dass $s \in L(\sigma)$ gilt. Im Schnitt können allerdings noch Spuren s mit größerem Inhalt enthalten sein, also Spuren s mit $cont(s) = cont(t) \cup \{\, \sigma_1 \,\}$ und $\sigma_1 \notin cont(t)$. Genau diese Spuren s liegen in der Vereinigung und werden entfernt. $\square$

Da $L(\sigma)$ boolsche Kombination von Sprachen der Form (3.5.1) ist, gilt dies auch für $\pi^{-1}\pi(t)$. Damit ist gezeigt, dass sich jede von $\diamond$ erkannte Sprache L als boolsche Kombination von Sprachen der Form (3.5.1) darstellen lässt.

4 Anwendungen

Wir wollen nun einige Sprachklassen näher betrachten. Das Schützenberger-Produkt kann hier in Beweisen von Inklusionsbeziehungen als Werkzeug eingesetzt werden. Es sei M ein Monoid. Die Familie der sternfreien Sprachen $SF(M)$, ist die kleinste Familie von Teilmengen von M, für die gilt

- $\emptyset \in SF(M)$
- $\{ a \} \in SF(M)$ für alle $a \in M$
- $A, B \in SF(M) \Rightarrow A \cup B \in SF(M)$,
  $A \cdot B = \{ ab \mid a \in A, b \in B \} \in SF(M)$,
  $\overline{A} \in SF(M)$

Die Familie der erkennbaren Mengen $Rec(M)$, ist die Familie der Teilmengen $A \subseteq M$, die von einem endlichen Monoid, bzw. einem endlichen Automaten erkannt werden. Wie wir sehen werden, gilt für Spurmonoide $\mathbb{M}$ die Beziehung $SF(\mathbb{M}) \subseteq Rec(\mathbb{M})$. Wir wollen die Menge der sternfreien Sprachen nun genauer betrachten. Hierzu benötigen wir *aperiodische* Monoide. Ein Monoid M heißt aperiodisch, falls eine Zahl $m \in \mathbb{N}$ existiert, so dass für alle $x \in M$ gilt $x^m = x^{m+1}$.

Lemma 4.1 *Sei $\mathbb{M}(\Sigma, I)$ ein Spurmonoid und $M_1, \ldots, M_\ell$ endliche, aperiodische Monoide. Dann ist das Schützenbergerprodukt $\diamond(\Sigma \cup \{\varepsilon\}, M_1, \ldots, \Sigma \cup \{\varepsilon\}, M_\ell)$ ebenfalls endliches, aperiodisches Monoid.*

Beweis: Offensichtlich ist $\diamond$ endliches Monoid. Es bleibt zu zeigen, dass $\diamond$ aperiodisch ist. Es muss also gezeigt werden, dass ein $m \in \mathbb{N}$ existiert, so dass für alle Mengen $A \in \diamond$ gilt $A^m = A^{m+1}$. Zur Erinnerung, $A \cdot B$ besteht aus allen Produkten $x \cdot y$ mit $x \in A$, $y \in B$ und $x \cdot y$ ist definiert. Wir betrachten zunächst A^*. Elemente aus A^* sind definierte Produkte der Form $x_1 \cdot \ldots \cdot x_n$ mit $n \in \mathbb{N}$ und $x_i \in A$ für alle $1 \leq i \leq n$. Wir fassen nun gleiche Elemente zusammen und erhalten so $x_1^{r_1} \cdot \ldots \cdot x_k^{r_k}$ mit $x_i \neq x_{i+1}$ und $r_i \in \mathbb{N}^+$. Wir behaupten, es ist $x_i \neq x_j$ für alle $1 \leq i \neq j \leq k$. Angenommen es gäbe $i \neq j$ mit $x_i = x_j$. Œ sei $i < j$. Da $x_i \neq x_{i+1}$ für alle $1 \leq i \leq k-1$ gilt, ist $j > i+1$. Seien $x, y \in A$ und x und y sind in allen verschiedenen Komponenten unabhängig. Dann folgt nach Definition des Schützenberger-Produkts $x = y$. Da $x_i \neq x_{i+1}$ müssen also mindestens zwei unterschiedliche Komponenten in x_i und x_{i+1} abhängig sein. Damit ist das Produkt undefiniert:

$$x_i = (b_1,\ y_1,\ \ldots\ \ldots\ \ldots\ b_\ell,\ x_\ell)$$

$$\lightning$$

$$x_{i+1} = (c_1,\ z_1,\ \ldots\ \ldots\ \ldots\ c_\ell,\ z_\ell)$$

oder

$$x_i = (b_1,\ y_1,\ \ldots\ \ldots\ \ldots\ b_\ell,\ x_\ell)$$

$$x_{i+1} = (c_1,\ z_1,\ \ldots\ \ldots\ \ldots\ c_\ell,\ z_\ell)$$

$$\vdots$$

$$\lightning$$

$$x_i = (b_1,\ y_1,\ \ldots\ \ldots\ \ldots\ b_\ell,\ x_\ell)$$

Es folgt die Behauptung $x_i \neq x_j$ für alle $1 \leq i \neq j \leq k$. Da die Monoide $M_1, \ldots, M_\ell$ aperiodisch sind, existieren Zahlen $n_1, \ldots, n_\ell \in \mathbb{N}$ mit $x^{n_i} = x^{n_i+1}$ für alle $x \in M_i$, $1 \leq i \leq \ell$. Setze jetzt $n := max\,\{\, n_1, \ldots, n_\ell \,\}$ und $\lambda := |\Sigma \cup \{\varepsilon\}| \cdot |M_1| \cdot \ldots \cdot |\Sigma \cup \{\varepsilon\}| \cdot |M_\ell|$. Es gibt höchtens λ viele verschiedene Elemente in A. Für Elemente $x_1^{r_1} \cdot \ldots \cdot x_k^{r_k}$ aus A^* gilt damit $k \leq \lambda$. Wir betrachten nun A^m für $m := n \cdot \lambda$. Es ist $A^m = A^{m+1}$:
Œ sei $m \geq 2$. Für $x = (b_1, y_1, \ldots, b_\ell, y_\ell) \in A$ folgt aus $x^m \in A^m$ oder $x^{m+1} \in A^{m+1}$, dass für alle $1 \leq i \leq \ell$ gilt $b_i = \varepsilon$. Es ist $x^m = (\varepsilon, y_1^m, \ldots, \varepsilon, y_\ell^m) = (\varepsilon, y_1^{m+1}, \ldots, \varepsilon, y_\ell^{m+1}) = x^{m+1}$. Es folgt $x^m \in A^m \Leftrightarrow x^{m+1} \in A^{m+1}$. Betrachte nun beliebige Tupel $x_1^{r_1} \cdot \ldots \cdot x_k^{r_k}$ mit $r_1 + \ldots + r_k = m$ aus A^m. Wegen $k \leq \lambda$ und $m = n \cdot \lambda$ gilt für mindestens eine Hochzahl $r_i \geq n$. Dann ist $x_i^{r_i} = x_i^{r_i+1}$ und $x_1^{r_1} \cdot \ldots x_{i-1}^{r_{i-1}} x_i^{r_i+1} x_{i+1}^{r_{i+1}} \cdot \ldots \cdot x_k^{r_k} \in A^{m+1}$. Analog folgt $A^{m+1} \subseteq A^m$, also insgesamt $A^m = A^{m+1}$. □

4.1 Schützenbergers Theorem für sternfreie Sprachen

Satz 4.2 (Schützenberger) *Für Spursprachen $L \subseteq \mathbb{M}(\Sigma, I)$ über einem Unabhängigkeitsalphabet (Σ, I) sind folgende Aussagen äquivalent:*

(a) L wird von einem endlichen, aperiodischen Monoid erkannt

(b) $L \in SF(\mathbb{M})$

Schützenberger hat diese Aussage 1965 bewiesen. Der Beweis findet sich im Orginalartikel [17] oder in [15, Theorem 2.6.1]. Diese Aussage von Schützenberger wurde in [7] auf Spurmonoide übertragen. Wir wollen hier einen einfacheren Beweis der Aussage angeben.

Satz 4.3 *Es sei $L \in SF(\mathbb{M})$ eine sternfreie Spursprache über dem Spurmonoid $\mathbb{M}(\Sigma, I)$. Dann existiert ein endliches, aperiodisches Monoid das L erkennt.*

Beweis: Beweis mit Induktion über den Aufbau von sternfreien Spursprachen. Die Mengen $\emptyset$ und $\{\, s \,\}$ für $s \in \mathbb{M}$ lassen sich offensichtlich mit einem endlichen, aperiodischen Monoid erkennen. Seien $L_1, L_2 \in SF(\mathbb{M})$ und L_i wird vom endlichen, aperiodischen Monoid M_i erkannt, $i = 1, 2$. Dann erkennt das endliche, aperiodische Monoid $M_1 \times M_2$

die Sprache $L_1 \cup L_2$. Das Produkt der Sprachen $L_1 L_2$ wird vom Schützenberger-Produkt $\diamond(\Sigma \cup \{\varepsilon\}, M_1, \Sigma \cup \{\varepsilon\}, M_2)$ erkannt. Das Schützenberger-Produkt ist nach Lemma 4.1 ein endliches, aperiodisches Monoid. Sei nun $X \subseteq M$ die Teilmenge des Monoids M, die $L \in SF(\mathbb{M})$ mittels eines Homomorphismus φ erkennt. Dann erkennt $M \setminus X$ das Komplement von L, es gilt $\overline{L} = \varphi^{-1}(M \setminus X)$. □

Ich danke Volker Diekert und Manfred Kufleitner für die in diesem Abschitt folgenden Aussagen und Beweise, die von mir nur an einigen Stellen abgeändert wurden.

Satz 4.4 *Es sei $L \subseteq \mathbb{M}$ eine Spursprache über dem Spurmonoid $\mathbb{M}(\Sigma, I)$, die von einem endlichen, aperiodischen Monoid M erkannt wird. Dann ist L sternfrei, $L \in SF(\mathbb{M})$.*

Zunächst benötigen wir für den Beweis die universelle Eigenschaft freier Monoide.

Satz 4.5 (Universelle Eigenschaft freier Monoide) *Sei M ein Monoid und $\varphi : \Sigma \to M$ eine Abbildung. Dann lässt sich φ eindeutig zu einem Homomorphismus $\varphi^* : \Sigma^* \to M$ erweitern. Dazu definieren wir $\varphi^*(\varepsilon) := 1$ und $\varphi^*(a_1 \dots a_n) := \varphi(a_1) \dots \varphi(a_n)$ für $n \geq 1$.*

$$\begin{tikzcd}
\Sigma \arrow[r, "\varphi"] \arrow[d, hook] \arrow[dr, phantom, "///"] & M \\
\Sigma^* \arrow[ur, dashed, "!\exists\varphi^*"'] &
\end{tikzcd}$$

Beweis: Seien $a_1 \dots a_n$ und $b_1 \dots b_m$ zwei Wörter aus Σ^*. Dann gilt

$$\varphi^*(a_1 \dots a_n b_1 \dots b_m) = \varphi(a_1) \dots \varphi(a_n)\varphi(b_1) \dots \varphi(b_m) = \varphi^*(a_1 \dots a_n)\varphi^*(b_1 \dots b_m)$$

Also ist φ^* ein Monoid-Homomorphismus. Sei $\psi : \Sigma^* \to M$ ein beliebiger Monoid-Homomorphismus mit $\psi(a) = \varphi(a)$ für alle $a \in \Sigma$. Dann ist $\psi(a_1 \dots a_n) = \psi(a_1) \dots \psi(a_n) = \varphi(a_1) \dots \varphi(a_n) = \varphi^*(a_1 \dots a_n)$. Also ist $\psi = \varphi^*$ und φ^* ist eindeutig. □

Wir besprechen nun eine Konstruktion von Monoiden, welche es uns erlauben wird, im folgenden Beweis eine Induktion nach der Größe von Monoiden durchzuführen. Sei M ein Monoid und $c \in M$. Wir definieren eine Verknüpfung $\circ$ auf $cM \cap Mc$ durch

$$xc \circ cy = xcy$$

Sei $xc = x'c$ und $cy = cy'$. Dann gilt $xc \circ cy = xcy = x'cy = x'cy' = x'c \circ cy'$. Dies zeigt, dass $\circ$ wohldefiniert ist. Seien $cx = x'c$ und cy Elemente in $cM \cap Mc$. Dann gilt $cx \circ cy = x'c \circ cy = x'cy = cxy$. Die Rechenregeln von M übertragen sich hierdurch auf $cM \cap Mc$. Insbesondere ist die Verknüpfung $\circ$ auf $cM \cap Mc$ assoziativ, da die Verknüpfung in M assoziativ ist. Damit ist $(cM \cap Mc, \circ, c)$ ein Monoid mit neutralem Element c. Wir nehmen im Folgenden stets an, dass M endlich und aperiodisch ist und dass $x^n = x^{n+1}$ gilt für alle $x \in M$. Wie bei der Assoziativität folgt $x^n = x^{n+1}$ für alle $x \in cM \cap Mc$. Wir benötigen die folgende Eigenschaft von aperiodischen Monoiden.

Lemma 4.6 *Sei M aperiodisches Monoid. Dann gilt $x_1 \dots x_k = 1$ für $k \geq 1$ in M genau dann, wenn $1 = x_i$ für alle $1 \leq i \leq k$.*

Beweis: Es genügt, die Behauptung für $k = 2$ zu zeigen. Sei $y^n = y^{n+1}$ für alle $y \in M$. Aus $1 = xy$ folgt

$$1 = x \cdot 1 \cdot y = x^2 y^2 = x^n y^n = x^{n+1} y^n = x \cdot 1 = x$$

Analog zeigt man $1 = y$. Die Umkehrung ist trivial. □

Beweis (Satz 4.4): Sei $\varphi : \mathbb{M} \to M$ ein Homomorphismus, der L erkennt. Es gilt $L = \bigcup_{x \in \varphi(L)} \varphi^{-1}(x)$. Deshalb genügt es zu zeigen, dass $\varphi^{-1}(x)$ sternfrei ist. Sei $\Sigma_1 = \{ a \in \Sigma \mid \varphi(a) = 1 \}$. Für $A \subseteq \Sigma$ ist $A^* = \{ s \in \mathbb{M} \mid alph(s) \subseteq A \} \subseteq \mathbb{M}$ sternfrei. Mit Lemma 4.6 folgt $\varphi^{-1}(1) = \Sigma_1^* \in SF(\mathbb{M})$. Sei nun $x \neq 1$. Für $c \in \Sigma$ mit $\varphi(c) \neq 1$ definieren wir $\Sigma_c = \Sigma \setminus \{ c \}$ und $X = \{ \varphi(s) \mid s \in \Sigma_c^* \}$. Durch Einschränkung von φ auf den Definitionsbereich Σ_c^* ergibt sich nun der Homomorphismus $\varphi_c : \Sigma_c^* \to X : s \mapsto \varphi(s)$. Da jede Spur s mit $\varphi(s) \neq 1$ ein Zeichen c enthält mit $\varphi(c) \neq 1$, gilt

$$\varphi^{-1}(x) = \bigcup_{\substack{c \in \Sigma \\ \varphi(c) \neq 1}} \bigcup_{x = x_1 x_2 x_3} \varphi_c^{-1}(x_1) \cdot (\varphi^{-1}(x_2) \cap \mathbb{M}_{c-} \cap \mathbb{M}_{-c}) \cdot \varphi_c^{-1}(x_3)$$

mit

$$\begin{aligned} \mathbb{M}_{c-} &:= \{ t \in \mathbb{M} \mid \exists t_1 \in \mathbb{M} : t = ct_1 \text{ und } \forall a \neq c \nexists t_2 : at_2 = t \} \\ \mathbb{M}_{-c} &:= \{ t \in \mathbb{M} \mid \exists t_1 \in \mathbb{M} : t = t_1 c \text{ und } \forall a \neq c \nexists t_2 : t_2 a = t \} \end{aligned}$$

Die Sprache $\varphi_c^{-1}(x_1)$ enthält den größt möglichen Teil vor dem ersten Vorkommen von c. Die Sprache $\varphi_c^{-1}(x_3)$ entspricht dem größt möglichen Teil nach dem letzten Vorkommen von c. Mit Induktion nach der Größe des Alphabets folgt, dass $\varphi_c^{-1}(x_1)$ und $\varphi_c^{-1}(x_3)$ sternfrei sind. Es verbleibt zu zeigen, dass für $x \in \varphi(c)M \cap M\varphi(c)$ die Sprache $\varphi^{-1}(x) \cap \mathbb{M}_{c-} \cap \mathbb{M}_{-c}$ sternfrei ist. Sei $<$ eine lineare Ordnung auf dem Alphabet Σ. Für jede Spur $t \in \mathbb{M}$ bezeichnet $\hat{t} \in \Sigma^*$ die lexikographische Normalform von t, also das minimale Wort in Bezug auf die lexikographische Ordnung, das t repräsent. Für ein $t \in \{ \varepsilon \} \cup \mathbb{M}_{c-}$ sei $\hat{t} = cv_1 cv_2 \dots cv_k$ die lexikographische Normalform. Wir definieren damit eine Substitution:

$$\sigma : \{ \varepsilon \} \cup \mathbb{M}_{c-} \to X^* : t \mapsto \varphi_c(v_1) \dots \varphi_c(v_k)$$

wobei $v_i \in \Sigma_c^*$ gilt. Hierbei bezeichnet X^* das freie Monoid über der Menge X, welche auch ein Untermonoid von M bildet. Sei $\psi : X^* \to \varphi(c)M \cap M\varphi(c)$ der Homomorphismus, der sich aus $X \to \varphi(c)M \cap M\varphi(c) : \varphi_c(s) \mapsto \varphi(csc)$ mit Lemma 4.5 ergibt. Für $s = cv_1 \dots cv_k \in \{ \varepsilon \} \cup \mathbb{M}_{c-}$ mit $v_i \in \Sigma_c^*$ und $k \geq 0$ gilt nun

$$\begin{aligned} \psi\sigma(s) &= \psi(\varphi_c(v_1)\varphi_c(v_2) \dots \varphi_c(v_k)) \\ &= \varphi(cv_1c) \circ \varphi(cv_2c) \circ \dots \circ \varphi(cv_kc) \\ &= \varphi(cv_1cv_2 \dots cv_kc) \\ &= \varphi(sc) \end{aligned}$$

Man beachte, dass formal $\psi\sigma(s) \in \varphi(c)M \cap M\varphi(c)$ gilt, wohingegen $\varphi(sc) \in M$ ist. Da $\varphi(c)M \cap M\varphi(c) \subseteq M$ gilt, macht dies trotzdem Sinn. Es folgt

$$sc \in \varphi^{-1}(x) \cap \mathbb{M}_{c-} \cap \mathbb{M}_{-c} \Leftrightarrow s \in \sigma^{-1}\psi^{-1}(x)$$

Deshalb genügt es $\sigma^{-1}\psi^{-1}(x) \in SF(\mathbb{M})$ zu zeigen. Mit Induktion nach der Größe des erkennenden Monoids folgt $\psi^{-1}(x) \in SF(X^*)$, und mit Induktion nach der Größe des Alphabets gilt $\varphi_c^{-1}(x) \in SF(\mathbb{M})$ für alle $x \in X$. Mit struktureller Induktion zeigen wir $\sigma^{-1}(K) \in SF(\mathbb{M})$ für jede Sprache $K \in SF(X^*)$:

$$\begin{aligned}
\sigma^{-1}(x) &= c \cdot \varphi_c^{-1}(x) && \text{für } x \in X \\
\sigma^{-1}(X^* \setminus K) &= (\{\,\varepsilon\,\} \cup \mathbb{M}_{c-}) \setminus \sigma^{-1}(K) && \text{für } K \in SF(X^*) \\
\sigma^{-1}(K_1 \cup K_2) &= \sigma^{-1}(K_1) \cup \sigma^{-1}(K_2) && \text{für } K_1, K_2 \in SF(X^*) \\
\sigma^{-1}(K_1 \cdot K_2) &= \sigma^{-1}(K_1) \cdot \sigma^{-1}(K_2) && \text{für } K_1, K_2 \in SF(X^*)
\end{aligned}$$

Die letzte Beziehung gilt wegen $\sigma(cs_1cs_2) = \sigma(cs_1)\sigma(cs_2)$.

□

4.2 Erkennbare Ausdrücke

Ochmanski hat in [12] sogenannte erkennbare Ausdrücke $REC(\mathbb{M})$ zur Beschreibung der erkennbaren Sprachen $Rec(\mathbb{M})$ entwickelt. Um dies zu erklären, benötigen wir einige Definitionen. Eine Spur $s \in \mathbb{M}$ heißt genau dann *zusammenhängend*, wenn keine Zerlegung $s = s_1s_2$ mit $s_1, s_2 \neq \varepsilon$ und $alph(s_1) \times alph(s_2) \subseteq I$ existiert. Für Sprachen $L \subseteq \mathbb{M}$ definieren wir die Menge $Con(L)$ als

$$Con(L) = \{\, t \in \mathbb{M} \mid t \text{ ist zusammenhängend und } \exists s_1, s_2 \in \mathbb{M} : s_1ts_2 \in L \,\}$$

Die *nebenläufige Iteration „co“* ist definiert als

$$L^{co} = (Con(L))^*$$

Damit können wir die Menge der *erkennbaren Ausdrücke* $REC(M)$ definieren als kleinste Familie von Teilmengen von M mit

- $\emptyset \in REC(M)$
- $\{\, a \,\} \in REC(M)$ für alle $a \in M$
- $A, B \in REC(M) \Rightarrow (A \cup B) \in REC(M)$,
  $(A \cdot B) \in REC(M)$,
  $(A^{co}) \in REC(M)$

Wir erhalten aus einem erkennbaren Ausdruck $E \in REC(M)$ eine Sprache $L(E)$, indem wir die Zeichen $\cup$ und $\cdot$ wie gewohnt interpretieren sowie $L(E^{co}) := L(E)^{co}$ definieren.

Satz 4.7 *Sei $E \in REC(\mathbb{M})$ ein erkennbarer Ausdruck über dem Spurmonoid $\mathbb{M}(\Sigma, I)$. Dann ist die Sprache $L(E)$ erkennbar.*

Ein Beweis der Aussage findet sich in [2, Kapitel 2.2]. Wir wollen diesen Beweis mit Hilfe des Schützenberger-Produkts vereinfachen.

Lemma 4.8 *[2, Prop. 2.2.4] Sei $L \in Rec(\mathbb{M})$ eine erkennbare Spursprache, dann ist auch $Con(L) \in Rec(\mathbb{M})$ eine erkennbare Spursprache.*

Lemma 4.9 *[2, Prop. 2.2.6] Sei $L \in Rec(\mathbb{M})$ eine erkennbare Sprache, die nur aus zusammenhängenden Spuren besteht, also $L \setminus \{1\} = Con(L)$. Dann ist die Sprache L^* erkennbar, $L^* \in Rec(\mathbb{M})$.*

Die Beweise dieser beiden Lemmata finden sich in [2] und werden hier nicht wiedergegeben.

Beweis (Satz 4.7): Beweis durch strukturelle Induktion. Für $E = \emptyset$ und $E = \{a\}$ ist $L(E)$ offensichtlich erkennbar. Seien E_1, E_2 erkennbare Ausdrücke und M_1, M_2 endliche Monoide, die $L(E_1)$ bzw. $L(E_2)$ erkennen. Dann erkennt das endliche Monoid $M_1 \times M_2$ die Vereinigung $L(E_1 \cup E_2) = L(E_1) \cup L(E_2)$. Das Schützenberger-Produkt $\diamond(\Sigma \cup \{\varepsilon\}, M_1, \Sigma \cup \{\varepsilon\}, M_2)$ erkennt $L(E_1 \cdot E_2) = L(E_1) \cdot L(E_2)$. Es bleibt zu zeigen, dass $L(E_1)^{co} = (Con(L(E_1)))^*$ eine erkennbare Sprache ist. Die Sprache $Con(L(E_1))$ ist nach Lemma 4.8 erkennbar. Nach Lemma 4.9 ist auch $L(E_1)^{co} = (Con(L(E_1)))^*$ erkennbar. □

Es gilt also $\{ L(E) \mid E \in REC(\mathbb{M}) \} \subseteq Rec(\mathbb{M})$. Ochmanskis Theorem besagt, dass auch die Umkehrung gilt.

Satz 4.10 (Ochmanski, [12, Thm. 9]) *Sei $\mathbb{M}(\Sigma, I)$ ein Spurmonoid und $L \subseteq \mathbb{M}$ eine Spursprache. Dann ist L genau dann erkennbar, $L \in Rec(\mathbb{M})$, wenn ein erkennbarer Ausdruck $E \in REC(\mathbb{M})$ mit $L(E) = L$ existiert.*

Ein Beweis dieser Aussage findet sich in [2, Kapitel 2.3]. Insgesamt gilt damit die Beziehung $\{ L(E) \mid E \in REC(\mathbb{M}) \} = Rec(\mathbb{M})$.

5 Zusammenfassung

In der vorliegenden Studienarbeit wurde das Schützenberger-Produkt endlicher Monoide auf endliche I-Monoide erweitert. Im 2. Kapitel wurden zunächst grundlegende Definitionen angegeben. Das Schützenberger-Produkt $\diamond$ endlicher Monoide $M_1, \ldots, M_\ell$ ist als Menge von $(\ell+1) \times (\ell+1)$-Matrizen definiert [17, 19, 16, 14]. Werden Sprachen L_i über dem Alphabet Σ von den endlichen Monoiden $M_1, \ldots, M_\ell$ erkannt, so erkennt das Schützenberger-Produkt genau boolsche Kombinationen von markierten Produkten der Form $L_{i_0} a_1 L_{i_1} \ldots a_r L_{i_r}$ mit $0 \leq r \leq \ell$, $0 \leq i_0 < i_1 < \ldots < i_r \leq \ell$, $a_k \in \Sigma$. Als nächstes wurde das von Mazurkiewicz stammende Konzept der Spuren eingeführt [11]. Das Spurmonoide ist definiert als $\mathbb{M}(\Sigma, I) = \Sigma^* / \{ u, v \in \Sigma^* : uv = vu \mid (u, v) \in I \}$ über einem Alphabet Σ und einer symmetrischen, irreflexiven Relation $I \subseteq \Sigma \times \Sigma$. Das Paar (Σ, I) heißt Unabhängigkeitsalphabet. Es ist $(x, y) \in I$ für $x, y \in \Sigma^*$ genau dann, wenn u durch endlich oftes Vertauschen unabhängiger Zeichen in v überführt werden kann. Unabhängige Zeichen, also Zeichen $a, b \in \Sigma$ mit $(a, b) \in I$, können als vertauschbare Prozesse in nebenläufigen Systemen interpretiert werden. Spursprachen $L \subseteq \mathbb{M}(\Sigma, I)$ eignen sich damit zur Beschreibung solcher nebenläufigen Systeme. Wortsprachen können mittels endlichen Monoiden beschrieben werden. Für Spursprachen hat sich diese Methode bisher als nicht sonderlich erfolgreich herausgestellt. Daher haben Kufleitner und Weil Unabhängigkeitsmonoide (M, I) zur Beschreibung von Spursprachen entwickelt [10]. Dies sind Monoide M, versehen mit einer symmetrischen Relation $I \subseteq M \times M$, die noch einige weitere Eigenschaften erfüllen muss. Starke I-Homomorphismen $\varphi : (M, I) \to (N, J)$ sind Homomorphismen zwischen I-Monoiden (M, I) und (N, J), wobei Elemente in M genau dann unabhängig sind, wenn die Bilder der Elemente unter φ abhängig sind in N. Um im weiteren Verlauf der Arbeit Elemente verschiedener I-Monoide auf Unabhängigkeit zu vergleichen, wurde das Skeletonmonoid $\mathbb{S}$ eingeführt. Ein Skeletonmonoid repräsentiert ein Unabhängigkeitsalphabet in kompakter Form.

In Kapitel 3 wird das Schützenberger-Produkt $\diamond$ auf I-Monoide erweitert. Für ein Unabhängigkeitsalphabet (Σ, I) und I-Monoide $M_1, \ldots, M_\ell$ wird es definiert als Teilmenge von $2^{\Sigma \cup \{\varepsilon\} \times M_1 \times \ldots \times \Sigma \cup \{\varepsilon\} \times M_\ell}$. Es ist $A \in \diamond$ genau dann, wenn gilt:

- Für alle $1 \leq i \leq \ell$ existiert in A genau ein Tupel $(1, \ldots, 1, x_i, 1, \ldots, 1)$
- Für alle $x, y \in A$ gilt $\mathbb{S}(x) = \mathbb{S}(y)$
- Seien $x = (b_1, x_1, \ldots, b_\ell, x_\ell), y = (c_1, y_1, \ldots, c_\ell, y_\ell) \in A$ mit x und y sind unabhängig in allen verschiedenen Komponenten. Dann folgt $x_i = y_i$ und $b_i = c_i$ für alle $1 \leq i \leq \ell$.

Zwei Elemente aus $\diamond$ können komponentenweise multipliziert werden, wenn gewisse Unabhängigkeitsbedingungen erfüllt sind. Das Produkt $x \cdot y$ für zwei Elemente $x = (b_1, x_1, \ldots, b_\ell, x_\ell), y = (c_1, y_1, \ldots, c_\ell, y_\ell) \in \diamond$ ist genau dann definiert, wenn alle Komponenten y_i zu den Komponenten x_j und b_j mit $1 \leq i < j \leq \ell$ unabhängig sind. Die Komponenten y_i müssen sich also an allen x_j und b_j vorbeischieben lassen. Analoges muss für die Marker c_i gelten. Im nächsten Schritt wird ein Satz bewiesen der besagt, dass Sprachen L der Form $L = a_1 L_1 \ldots a_k L_k$ vom erweiterten Schützenberger-Produkt $\diamond(\Sigma \cup \{\varepsilon\}, M_1, \ldots, \Sigma \cup \{\varepsilon\}, M_\ell)$ erkannt werden. Dabei sind L_i Spursprachen, die von den Monoiden M_i mittels starker I-Homomorphismen erkannt werden, $1 \leq i \leq \ell$. M_i ist I-Monoid, die Unabhängigkeitsrelation auf M_i ergibt sich aus der Unabhängigkeitsrelation auf Σ. Es wird die Konstruktion eines I-Homomorphismus $\varphi : \mathbb{M}(\Sigma, I) \to \diamond$ angegeben, der die Spursprache L erkennt. In einem weiteren Schritt wird gezeigt, dass sich jede Sprache L die von $\diamond$ mittels eines Homomorphismus φ erkannt, als boolsche Kombination von Sprachen der Form $a_1 L_1 \ldots a_k L_k$ darstellen lässt. Dabei ist $0 \leq k \leq |\Sigma \cup \{\varepsilon\}| + |M_1| + |\Sigma \cup \{\varepsilon\}| + |M_2| + \ldots + |M_{\ell-1}| + |\Sigma \cup \{\varepsilon\}|$ und $a_i \in \Sigma$ für alle $1 \leq i \leq k$. Des weiteren ist L_i für alle i eine von $M_1 \times M_2 \times \ldots \times M_\ell$ erkannte Sprache. Für den Beweis der Aussage werden die Elemente $x \in \varphi(t)$ einer Spur t näher betrachtet. Ein Element $x \in \varphi(t) = \varphi(b_1) \ldots \varphi(b_n)$ für eine Spur $t = b_1 \ldots b_n$ ergibt sich als Produkt von Elementen $\varphi(b_i)_{p_i} \in \varphi(b_i)$, $1 \leq i \leq n$. Es existiert unabhängig von der Spur $t \in \mathbb{M}$ eine maximale Anzahl m so, dass in jedem definierten Produkt $\varphi(b_1)_{p_1} \ldots \varphi(b_n)_{p_n}$ höchstens m Elemente nicht die Form $(1, y_1, \ldots, 1, y_\ell)$ mit unabhängigen Komponenten y_j haben. Dies ist eine wichtige Erkenntnis für den Beweis der Aussage, denn so lässt sich die Anzahl an Markern a_i beschränken, die zur Darstellung von L benötigt werden.

Im Kapitel 4 werden Anwendungen des erweiterten Schützenberger-Produkts aufgezeigt. Zunächst wird das Schützenberger Theorem für sternfreie Sprachen $SF(\mathbb{M})$ betrachtet. Es besagt, dass die von endlichen, aperiodischen Monoiden erkannten Sprachen genau den sternfreien Sprachen entsprechen. Da das Schützenberger-Produkt von aperiodischen I-Monoiden ein aperiodisches I-Monoid ist, lässt sich eine Richtung der Aussage von Schützenberger zu sternfreien Sprachen mit dem erweiterten Schützenberger-Produkt recht einfach beweisen. Für die Gegenrichtung wird ebenfalls ein zu bisherigen Versionen vereinfachter, vollständiger Beweis angegeben. Eine zweite Anwendung sind erkennbare Ausdrücke $REC(\mathbb{M})$. Diese Ausdrücke wurden von Ochmanski entwickelt um erkennbare Sprachen $Rec(\mathbb{M})$, also Sprachen die von endlichen Monoiden erkannt werden, zu beschreiben. Einem erkennbaren Ausdruck $E \in REC(\mathbb{M})$ lässt sich eine Sprache $L(E)$ zuordnen. Jede solche Sprache $L(E)$ wird von einem endlichen Monoid erkannt. Um dies mittels struktureller Induktion zu beweisen wird die Aussage benötigt, dass das Produkt zweier erkennbarer Sprachen $L(E_1) \cdot L(E_2)$ wieder erkennbar ist. Dies folgt direkt aus dem Schützenberger-Produkt. Da auch die Gegenrichtung gilt, beschreiben die erkennbaren Ausdrücken genau die erkennbaren Sprachen.

Literaturverzeichnis

[1] Bruyère, Véronique ; De Felice, Clelia ; Guaiana, Giovanna: On some decision problems for trace codings. In: *Theor. Comput. Sci.* 148 (1995), Nr. 2, S. 227–260. – ISSN 0304–3975 10

[2] Diekert, Volker: *Combinatorics on traces*. New York, NY, USA : Springer-Verlag New York, Inc., 1990. – ISBN 0–387–53031–2 30

[3] Diekert, Volker ; Rozenberg, Grzegorz (Hrsg.): *The Book of Traces*. River Edge, NJ, USA : World Scientific Publishing Co., Inc., 1995. – ISBN 9810220588

[4] Diekert, Volker ; Gastin, Paul ; Kufleitner, Manfred: A Survey on Small Fragments of First-Order Logic over Finite Words. In: *International Journal of Foundations of Computer Science* 19 (June 2008), Nr. 3, S. 513–548. – Special issue DLT 2007 5

[5] Diekert, Volker ; Métivier, Yves: Partial Commutation and Traces. In: Rozenberg, Grzegorz (Hrsg.) ; Salomaa, Arto (Hrsg.): *Handbook of Formal Languages: Volume 3. Beyond Words*. 1. Springer, Mai 2002, S. S. 457–534. – ISBN 9783540606499 17

[6] Eilenberg, Samuel: *Automata, Languages, and Machines*. Orlando, FL, USA : Academic Press, Inc., 1976. – ISBN 0122340027

[7] Guaiana, Giovanna ; Restivo, Antonio ; Salemi, Sergio: Star-free trace languages. In: *Theor. Comput. Sci.* 97 (1992), Nr. 2, S. 301–311. – ISSN 0304–3975 26

[8] Kirsten, Daniel: A Connection between the Star Problem and the Finite Power Property in Trace Monoids. In: *ICALP*, 1999, S. 473–482

[9] König, Roman: Graphs and free partially commutative monoids. In: *Theor. Comput. Sci.* 78 (1991), Nr. 2, S. 319–346. – ISSN 0304–3975 10

[10] Kufleitner, Manfred ; Weil, Pascal: *Independance Monoids*. Juni 2007 5, 8, 9, 10, 31

[11] Mazurkiewicz, Antoni: Concurrent program schemes and their interpretations / Aarhus University. 1977 (DAIMI-PB-78). – Forschungsbericht 5, 8, 31

[12] Ochmanski, Edward: Regular behaviour of concurrent systems. In: *Bulletin of the EATCS* 27 (1985), S. 56–67 29, 30

[13] Ochmanski, Edward ; Stawikowska, Krystyna: A Star Operation for Star-Free Trace Languages. In: *Developments in Language Theory*, 2007, S. 337–345

[14] Pin, Jean E. ; Miller, Raymond E. (Hrsg.): *Varieties Of Formal Languages*. Plenum Publishing Co., 1986. – ISBN 0306422948 7, 31

[15] Pippenger, Nicholas: *Theories of Computability*. Cambridge, England : Cambridge University Press, 1997. – ISBN 052155380–6 5, 26

[16] Reutenauer, Christophe: Sur les variétés de langages et de monoídes. In: *Proceedings of the 4th GI-Conference on Theoretical Computer Science*. London, UK : Springer-Verlag, 1979. – ISBN 3–540–09118–1, S. 260–265 7, 31

[17] Schützenberger, Marcel P.: On Finite Monoids Having Only Trivial Subgroups. In: *Information and Control* 8 (1965), Nr. 2, S. 190–194 5, 7, 26, 31

[18] Simon, Imre: The product of rational languages. In: *Proceedings of ICALP 1993, Lecture Notes in Computer Science 700*, 1993, S. 430–444 7

[19] Straubing, Howard: A generalization of the Schützenberger product of finite monoids. In: *Theor. Comput. Sci.* 13 (1981), Nr. 2, S. 137–150 7, 31

Erklärung

Hiermit versichere ich, diese Arbeit
selbständig verfasst und nur die
angegebenen Quellen benutzt zu haben.

(Bastian Bischoff)